Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1984-06

A computer program to calculate the supersonic flow over a solid cone in air or water.

Hughes, Patrick William

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/19209

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NPS-67-84-007

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A COMPUTER PROGRAM TO CALCULATE THE SUPERSONIC FLOW OVER A SOLID CONE IN AIR OR WATER

by

Patrick William Hughes
June 1984

Thesis Advisor: Allen E. Fuhs

Approved for public release; distribution unlimited.

Prepared for: Mr. Donald Phillips
Code R10A
Naval Surface Weapons Center, White Oak
Silver Spring, Maryland 20910

T222113

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Shumaker Superintendent	David Schrady Provost

This thesis was prepared in conjunction with research supported in part by the Naval Surface Weapons Center, White Oak, Silver Spring, Maryland.

Reproduction of all or part of this report is authorized.

Released as a
Technical Report by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NPS-67-84-007

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)
A Computer Program to Calculate the
Supersonic Flow Over a Solid Cone in Air
or Water

5. TYPE OF REPORT & PERIOD COVERED
Master's thesis;
June 1984

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(*s*)
Patrick William Hughes

8. CONTRACT OR GRANT NUMBER(*s*)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Element 62633N
Task SF 33327

11. CONTROLLING OFFICE NAME AND ADDRESS
Mr. Donald Phillips, Code R10A
Naval Surface Weapons Center, White Oak
Silver Spring, Maryland 20910

12. REPORT DATE
June 1984

13. NUMBER OF PAGES
116

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)
Supersonic hydrodynamics
Shock waves
Conical flow

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

The computer program calculates the supersonic flow over a cone in air or water. The main objective is to calculate the cone semi-vertex angle given prescribed initial conditions. The program is written in structured FORTRAN and implements Busemann's graphical integration technique. Supersonic flow over a cone in water is useful as a good first approximation to the motion of the metal jet from an explosive shaped-charge fired underwater.

A typical result for supersonic flow over a cone in water is as follows: given an upstream temperature, 323.16° Kelvin; upstream pressure, 1 bar;

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

20. Abstract Continued.

shock angle, 20.0 degrees; and pressure behind the shock front, 5 kilobars, the cone semi-vertex angle is calculated to be 7.23 degrees.

Generally, pressures involved in water flow are much larger than for air flow, and the cone semi-vertex angles for water flow are smaller than for air flow.

Approved for public release; distribution unlimited.

A Computer Program to Calculate
the Supersonic Flow Over a
Solid Cone in Air or Water

by

Patrick W. Hughes
Lieutenant, United States Navy
B.S., University of Washington, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1984

Thesis
H [illegible]753
c.2

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

ABSTRACT

The computer program calculates the supersonic flow over a cone in air or water. The main objective is to calculate the cone semi-vertex angle given prescribed initial conditions. The program is written in structured FORTRAN and implements Busemann's graphical integration technique. Supersonic flow over a cone in water is useful as a good first approximation to the motion of the metal jet from an explosive shaped-charge fired underwater.

A typical result for supersonic flow over a cone in water is as follows: given an upstream temperature, 323.16 Kelvin; upstream pressure, 1 bar; shock angle, 20.0 degrees; and pressure behind the shock front, 5 kilobars, the cone semi-vertex angle is calculated to be 7.23 degrees.

Generally, pressures involved in water flow are much larger than for air flow, and the cone semi-vertex angles for water flow are smaller than for air flow.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

TABLE OF CONTENTS

I. INTRODUCTION 8
  A. DESCRIPTION OF THE PROBLEM 8
  B. METHODOLOGY 9
II. FUNDAMENTAL EQUATIONS 11
  A. BACKGROUND 11
  B. GENERAL EQUATIONS 12
  C. EQUATIONS SPECIFIC TO AIR 19
  D. EQUATIONS SPECIFIC TO WATER 22
III. DESCRIPTION OF THE COMPUTER PROGRAM 28
  A. PROGRAM LOGIC 28
  B. USER INSTRUCTIONS 33
IV. PROGRAM CALCULATION RESULTS 37
  A. PROGRAM RESULTS FOR THE CALCULATION OF CONICAL FLOW IN AIR 37
  B. PROGRAM RESULTS FOR THE CALCULATION OF CONICAL FLOW IN WATER 40
V. CONCLUSIONS 52
APPENDIX A: PROGRAM FLOWCHARTS 54
APPENDIX B: SAMPLE PRINTOUTS 76
APPENDIX C: PROGRAM LISTING 86
LIST OF REFERENCES 113
BIBLIOGRAPHY 114
INITIAL DISTRIBUTION LIST 115

LIST OF TABLES

I. Nomenclature 27

II. Program Results and Comparisons 38

III. Program Results and Comparisons (cont'd.) 39

IV. Properties of Sea Water at a Shock Front 41

V. Cone Semi-vertex Angle Variation with Temperature 45

VI. Main Program Variables 62

VII. Main Program Variables (cont'd.) 63

VIII. Main Program Variables (cont'd.) 64

IX. Main Program Variables (cont'd.) 65

X. Subroutine WSHOCK Variables 72

XI. Subroutine WATVEL Variables 75

LIST OF FIGURES

2.1 Shock Cone and Typical Streamline 14
2.2 Hodograph Image of Typical Streamline 15
2.3 Cone and Flow Geometry 17
2.4 Geometry of the Oblique Shock Front 22
3.1 Module Hierarchy 29
3.2 Graphical Construction of Cone Flow 32
4.1 Freestream Mach Number vs Shock Angle for Supersonic Flow in Water 49
4.2 Freestream Mach Number vs Surface Mach Number for Supersonic Flow in Water 50
4.3 Freestream Mach Number vs Drag Coefficient for Supersonic Flow in Water 51
A.1 Main Program Flowchart 55
A.2 Main Program Flowchart (cont'd.) 56
A.3 Main Program Flowchart (cont'd.) 57
A.4 Main Program Flowchart (cont'd.) 58
A.5 Main Program Flowchart (cont'd.) 59
A.6 Main Program Flowchart (cont'd.) 60
A.7 Main Program Flowchart (cont'd.) 61
A.8 Subroutine CEKINP Flowchart 66
A.9 Subroutine CEKINP Flowchart (cont'd.) 67
A.10 Subroutine CEKINP Flowchart (cont'd.) 68
A.11 Subroutine DEFANG Flowchart 69
A.12 Subroutine WSHOCK Flowchart 70
A.13 Subroutine WSHOCK Flowchart (cont'd.) 71
A.14 Subroutine WATVEL Flowchart 73
A.15 Subroutine WATVEL Flowchart (cont'd.) 74

I. INTRODUCTION

A. DESCRIPTION OF THE PROBLEM

The solution of the hydrodynamic equations describing supersonic flow over a cone in air has been well known since the 1930's. Until recently, the problem of describing the flow over a cone in water has been limited to solutions of the subsonic case. Primarily, calculations were limited to subsonic flow because researchers believed that supersonic flow in water was not feasible for normal vessels (such as a ship). For ordinary vessels in water, it is certainly true that supersonic flow past that vessel is highly improbable. However, the motion of the metal jet from an explosive shaped-charge fired underwater is supersonic.

This thesis presents a computer program which calculates the hydrodynamic flow past a cone in either water or air under supersonic conditions. The program utilizes the methods developed by previous researchers for calculating the supersonic flow in air and which have been suitably modified to describe the conditions in the water. Such modifications include utilizing the modified Tait equation, which is the "thermal" or "thermodynamic" equation of state for water, to describe the physical state of the water rather than the perfect gas law used for air.

In actuality, the cone liner in the jet from an explosive shaped-charge is blunt-nosed rather than conical. However, solution of the conical case is a preliminary requirement to solution of the actual blunt-nosed problem. The solution to the conical flow case will serve as an excellent test program for the solution to the blunt-nosed problem. This thesis presents a solution to the conical case and it is hoped that the program will assist the continuing research into the problems of utilizing explosive shaped-charges in an underwater environment.

E. METHODOLOGY

The computer program presented in this work was originally developed in the BASIC computer language using a Hewlett-Packard HP-67 computer. That program is the basis for this thesis. It was desired to translate the program into a higher-order computer language for execution on a large, mainframe computer system. This translation was desired in order to make the program more readily accessible to a wider body of researchers and in order to speed the execution time of the program. In this thesis, the following goals have been accomplished:

- Successfully translate the program from BASIC into a higher-order language. This goal was met by utilizing FORTRAN as the high-level language of choice. While FORTRAN has many drawbacks as a high-order language, it is still widely used in the scientific community. FORTRAN was used, therefore, so that the program will be useful to as wide an area of researchers as possible.

- Follow modern programming practices in the design and implementation of the program. As before, the choice of FORTRAN as the high-level language makes this goal somewhat more difficult. However, many computer scientists have demonstrated that structured programming practices can be achieved using FORTRAN. To the largest extent possible structured programming practices have been utilized.

- Present a "user-friendly", well-documented program. In this regard, literal use of comments occur in the program itself, meaningful variable names are used, and detailed flowcharts which demonstrate the logic of the program are included. In addition, due to limited interaction with the user, the user's responses are verified before the program executes.

II. FUNDAMENTAL EQUATIONS

A. BACKGROUND

The first step necessary to describe supersonic flow over a cone is to calculate the thermodynamic properties across a shock wave. The basic research into the change of thermodynamic properties across a shock wave in water was extensively conducted and reported upon in Underwater Explosion Research [Ref. 1] during and just after World War II. The best summarization of these works can be found in Cole [Ref. 2].

The primary source used as reference for the calculation of the hydrodynamic properties of sea water at the front of a shock wave is the work of Richardson, Arons, and Halverson [Ref. 3]. They utilized graphical techniques, which were rather crude and tedious, to calculate the thermodynamic data needed to describe the conditions of the sea water. Fuhs [Ref. 4] used the work of Richardson, et al., [Ref. 3], to develop a computer program for the HP41CV hand-held calculator which efficiently calculated the same thermodynamic properties. Fuhs' [Ref. 4] programs provided the basis for the FORTRAN subroutines, used in this work, which calculate the same thermodynamic properties such as pressure, temperature, and density.

The fundamental equations used to calculate the thermodynamic changes which occur across an oblique shock front in air have been well known for decades. These equations are exceptionally well described in Kinney and Graham [Ref. 5] and form the basis for the initial calculations for the supersonic flow past a cone in air.

The fundamental equation used to describe the thermodynamic state of air is the perfect gas law. In water, the modified Tait equation is the equation most often used to describe the thermodynamic state of the water. The modified Tait equation can be used to describe either pure water or sea water. The form of the modified Tait equation used in this thesis was taken from Holt [Ref. 6], who has continued to perform research in underwater explosion phenomena. An excellent discussion of the modified Tait equation and how it can be utilized is contained in Rowlinson [Ref. 7].

After the thermodynamic properties of the water (or air as the case may be) on the downstream side of the shock front have been calculated, an iteration method, utilizing an automated graphical construction first developed by Busemann, [Ref. 8], is used to progress from the shock front to the cone surface. The equations needed for use by this iteration method are fully described, for air, by Shapiro [Ref. 8], who describes their development and use. The methods which apply to the flow past a cone in air can, with the necessary changes made for the differences in the thermodynamics of the two fluids, be used to calculate the conical supersonic flow in water. These calculations form the basis for the main part of the FORTRAN program which follows. It should be noted here that Shapiro also points out the pioneering work of Taylor and Maccoll in the 1930's and 1940's on the methods of solution of the flow over a cone in air problem [Ref. 8].

B. GENERAL EQUATIONS

In the development of the equations which follow, it is assumed that these equations can be validly used to calculate the flow over the cone in either air or water. The equations were primarily developed by Shapiro [Ref. 8] for flow in air.

After calculating the thermodynamic changes which occur as a result of the shock front, it is assumed that the fluid properties will remain constant on imaginary cones having a common vertex. By this assumption, the flow past the cone can be calculated. The flow geometry is illustrated by figure 2.1 and figure 2.2, which show a typical streamline and its image in the hodograph plane.

As discussed thoroughly by Shapiro [Ref. 8], there is a discontinuous change in both direction and velocity across the oblique shock front. Points 1 and 2 (see figure 2.1 and figure 2.2) lie, therefore, on a common shock polar which originates at point 1. Between point 2 and the point 's', which is on the cone surface only at infinite distance, there is a region of conical flow where the stream properties vary continuously. The velocity vector to the point 's' in the hodograph plane defines what is called the cone semi-vertex angle with the centerline axis. In the methods which follow, the cone semi-vertex angle is the variable which is ultimately determined. Further, as pointed out by Shapiro [Ref. 8], since all streamlines in the flow experience the same entropy jump across the shock front, the flow between the shock front and the cone surface is both isentropic and irrotational.

The second-order differential equation which actually describes the flow of the fluid past the cone in air is fully developed by Shapiro [Ref. 8]. Shapiro notes that there are two methods commonly used to solve this equation. One method, developed first by Taylor and Maccoll [Ref. 8], performs a numerical integration of the equation. The second method, which integrates the equation using a graphical construction method, was first developed by Busemann. The program developed in this thesis utilizes Busemann's method, modified for performance on a modern high-speed computer, to perform the integration of the full second-order differential equation.

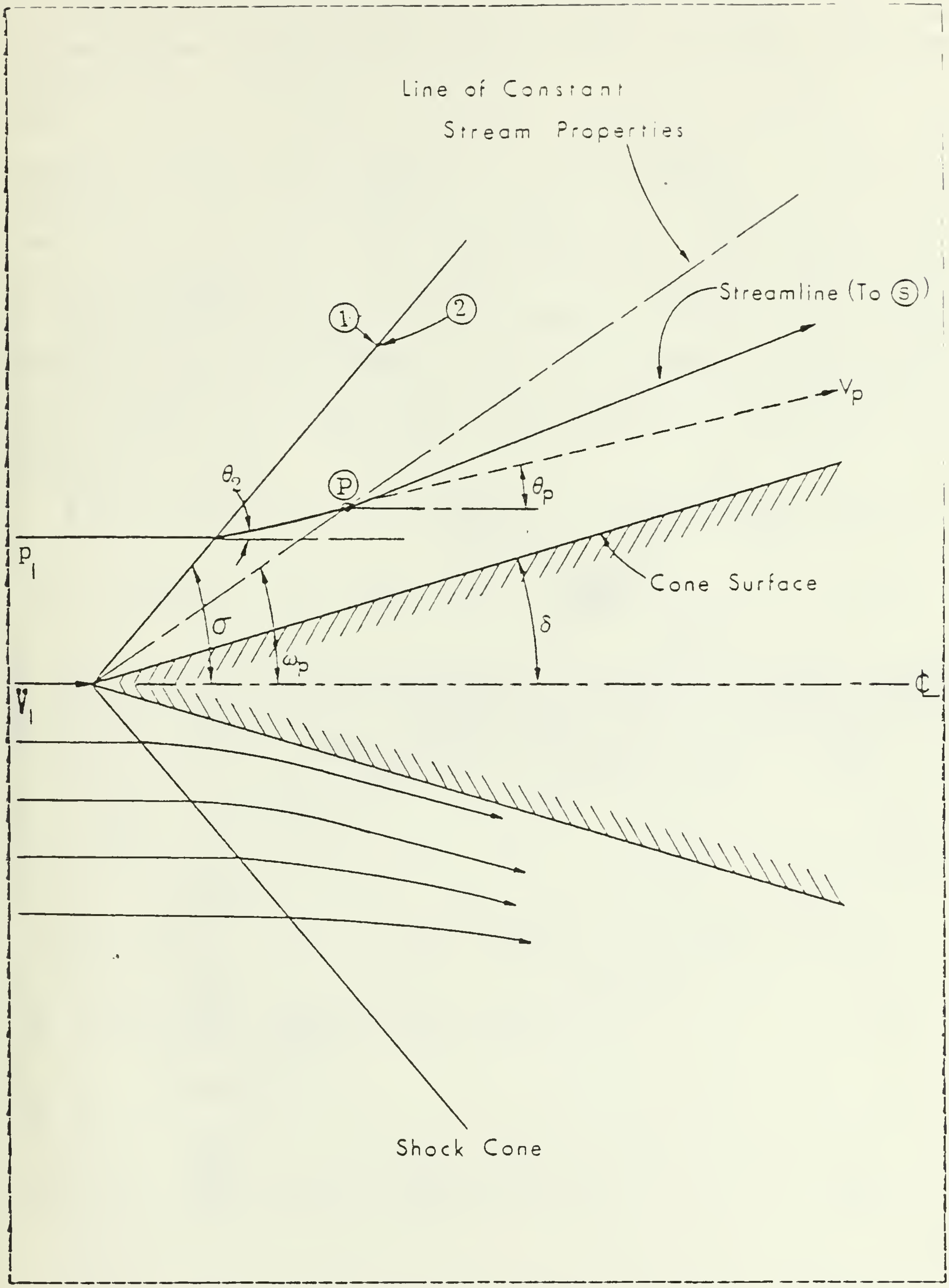

Figure 2.1 Shock Cone and Typical Streamline.

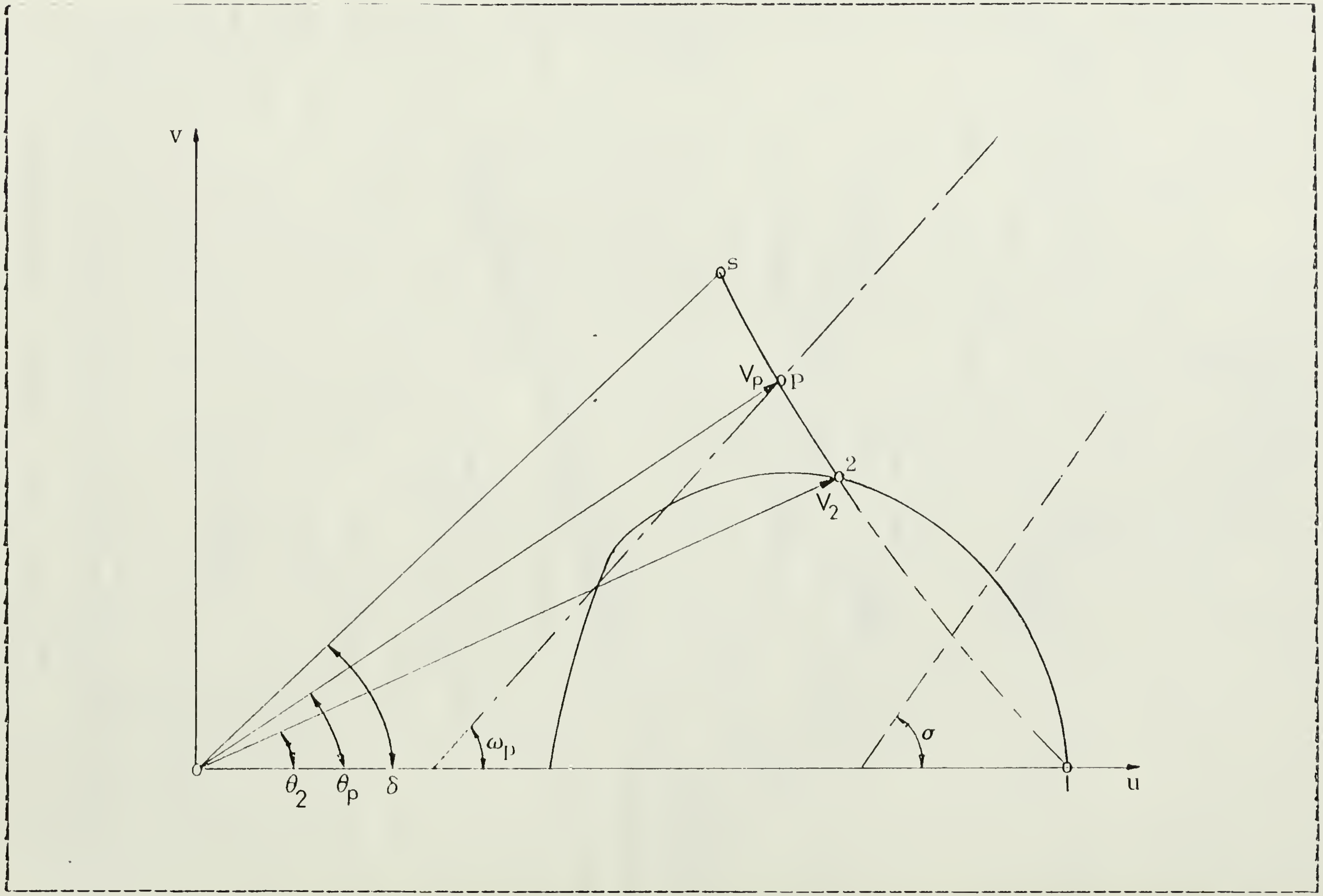

Figure 2.2 Hodograph Image of Typical Streamline.

In the following derivations, as per Shapiro [Ref. 8], the spherical coordinates r and ω have been used with the corresponding velocity components V_r and V_ω (see figure 2.3). Only the primary equations which are used in the computer program are presented in this thesis. A detailed derivation of the equations can be found in Shapiro [Ref. 8], and need not be repeated here. The nomenclature used in the equations developed in this chapter is detailed in Table I. In keeping with modern thought, the M-K-S (meter-kilogram-second) unit system has been used throughout this thesis except for occasional lapses during the water calculations when pressures are referred to in units of kilobars.

From the geometry of figure 2.3, it can be seen that:

$$V_r = V\cos(\omega - \theta) \quad \text{and} \quad V_\omega = -V\sin(\omega - \theta) \tag{2.1}$$

In the development of the actual second-order differential equation, Shapiro [Ref. 8] shows that, due to the condition of irrotationality, the following relation must be true:

$$V_\omega = \frac{dV_r}{d\omega} \tag{2.2}$$

Differentiating Eqn. 2.1 with respect to ω, one obtains:

$$\frac{dV_r}{d\omega} = -V\left(1 - \frac{d\theta}{d\omega}\right)\sin(\omega - \theta) + \frac{dV}{d\omega}\cos(\omega - \theta) \tag{2.3}$$

and

$$\frac{dV_\omega}{d\omega} = -V\left(1 - \frac{d\theta}{d\omega}\right)\cos(\omega - \theta) - \frac{dV}{d\omega}\sin(\omega - \theta) \tag{2.4}$$

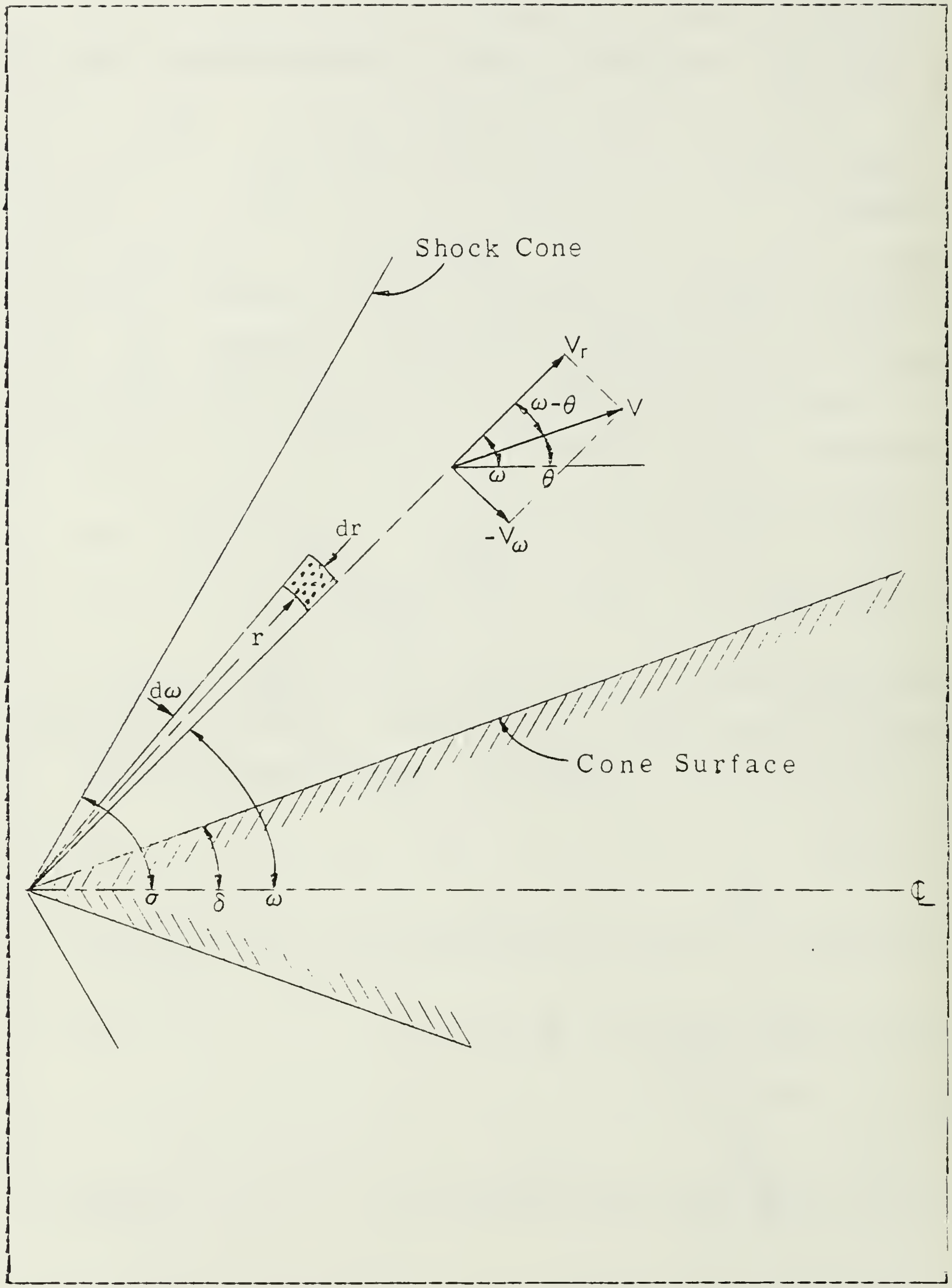

Figure 2.3 Cone and Flow Geometry.

But, from Eqn 2.1 and Eqn 2.2, it can be seen that:

$$\frac{dV_r}{d\omega} = V_\omega = -V\sin(\omega - \theta) \tag{2.5}$$

Substitution of this result into Eqn 2.3 yields:

$$\frac{d\theta}{d\omega} = \frac{-\frac{dV}{d\omega}}{V\tan(\omega - \theta)} \quad . \tag{2.6}$$

Shapiro [Ref. 8] shows that the equation governing the flow in the conical region is:

$$\begin{aligned}\left(\frac{k-1}{2}\right)\left(2V_r + V_\omega \cot\omega + \frac{dV_\omega}{d\omega}\right)\left(V_{max}^2 - V_\omega^2 - V_r^2\right) \\ = \left(V_r\frac{dV_r}{d\omega} + V_\omega\frac{dV_\omega}{d\omega}\right)V_\omega\end{aligned} \tag{2.7}$$

Eliminating $\frac{d\theta}{d\omega}$ from Eqn 2.4 and substituting the expressions for V_r, V_ω, $\frac{dV_r}{d\omega}$ and $\frac{dV_\omega}{d\omega}$, given by Eqns 2.1, 2.4 and 2.5, into Eqn 2.7 gives:

$$\frac{dV}{d\omega} = \frac{V\sin(\omega - \theta)\,\frac{\sin\theta}{\sin\omega}}{1 - \frac{2V^2\sin^2(\omega - \theta)}{(k-1)(V_{max}^2 - V^2)}} \quad . \tag{2.8}$$

Designating 'R' as the radius of curvature of the hodograph streamline, one obtains:

$$R = \frac{dV}{\sin(\omega - \theta)d\omega} \quad . \tag{2.9}$$

By combining Eqn 2.9 with Eqn 2.8, one arrives at:

$$R = \frac{\dfrac{\sin\theta}{V\ \sin\omega}}{1 - \dfrac{2V^2\ \sin^2(\omega - \theta)}{(k - 1)(V^2_{max} - V^2)}} \qquad (2.10)$$

But, the energy equation asserts that:

$$c^2 = (\frac{k - 1}{2})\ (V^2_{max} - V^2) \qquad (2.11)$$

therefore

$$R = \frac{\dfrac{\sin\theta}{V\ \sin\omega}}{1 - \dfrac{V^2\ \sin^2(\omega - \theta)}{c^2}} \qquad (2.12)$$

where c is the local speed of sound in the fluid.

Equation 2.12 is the basis for the calculation of the supersonic flow over the solid cone in either air or water. The graphical integration method invented by Busemann is adequately explained in Shapiro [Ref. 8] and need not be repeated here. Essentially, the computer program given in this work automates the Busemann graphical integration method for calculating the cone semi-vertex angle.

C. EQUATIONS SPECIFIC TO AIR

The equation of state for air is specified by the perfect gas law (under the assumption, that is, that the air behaves as a perfect gas). This law is quite elegant and simple and allows easy manipulation to obtain various quantities. The form of the perfect gas law most often used in the calculations of this thesis is:

$$p = \rho RT \qquad (2.13)$$

where R is the *specific* gas constant and is related to the universal gas constant, Λ, by:

$$R = \frac{\Lambda}{M_a} \tag{2.14}$$

In equation 2.14, M_a is the molecular weight of the air.

In air, the change in the thermodynamic properties of the gas as it crosses the shock front are easily calculated, as shown in Kinney and Graham [Ref. 5]. Since most fluid dynamics textbooks illustrate the development of the equations which follow, it is not necessary to derive them here. As mentioned previously, Kinney and Graham [Ref. 5] provide exceptionally lucid explanations and derivations. The principle equations used to calculate the thermodynamic changes which occur across the shock front in *air* are as follows:

$$\frac{\tan(\beta - \theta)}{\tan\beta} = \frac{2 + (k - 1)\, M_1^2 \sin^2\beta}{(k + 1)\, M_1^2 \sin^2\beta} \tag{2.15}$$

Equation 2.15 is used to iteratively determine the deflection angle θ. All other quantities in this equation are known (i.e. β is the shock angle and M_1 is the freestream Mach number, both of which are input parameters to the program for the air calculations).

$$[M_2 \sin(\beta - \theta)]^2 = \frac{2 + (k - 1)\, M_1^2 \sin^2\beta}{2kM_1^2 \sin^2\beta - (k - 1)} \tag{2.16}$$

Having determined θ from equation 2.15, equation 2.16 is used to determine the Mach number on the downstream side of the shock front.

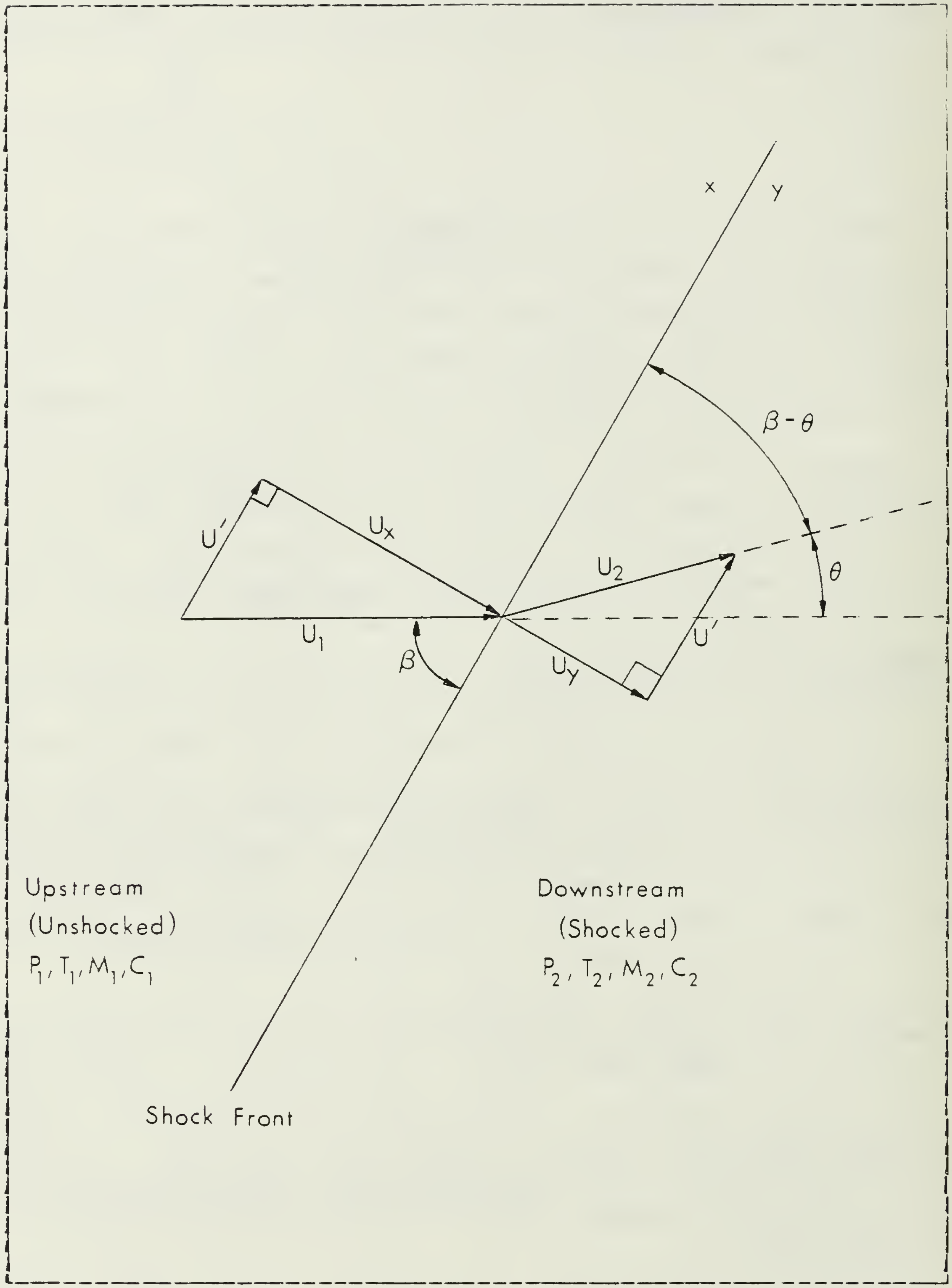

Figure 2.4 Geometry of the Oblique Shock Front.

D. EQUATIONS SPECIFIC TO WATER

In contrast to air, which has a relatively elegant and simple state equation, the equation of state for water is rather more complicated. The most commonly used state equation for water is known as the modified Tait equation, which may be written as:

$$p = B(S)\left[\left(\frac{\rho}{\rho}\right)^{n} - 1\right] \qquad (2.21)$$

where B(S) is a slowly varying function of entropy alone, n is approximately a constant equal to 7.15, and $\bar{\rho}$ is the value of the density for zero pressure. The above equation is from Holt [Ref. 6], but, in different forms, it is also described quite extensively in Cole [Ref. 2], Richardson, *et. al.*, [Ref. 3] and Rowlinson [Ref. 7]. As mentioned in the Introduction to this work, the modified Tait equation is the "thermal" equation of state for water. Cole [Ref. 2] shows that the modified Tait equation is of the form

$$p = B(S)\ \left[\left(\frac{v(T,0)}{v(T,p)}\right)^{n} - 1\right] \qquad (2.22)$$

or, in simpler terms,

$$p = p(v,T) \qquad (2.23)$$

Since the full modified Tait equation relates the three thermodynamic quantities of p, v, and T, it is called the "thermodynamic" or "thermal" equation of state.

Similarly, in contrast to air, the thermodynamic changes which occur as a result of the shock process in water cannot be easily delineated by simple equations as in the air case. However, a simplification can be made in the water case because, unlike in the air case, the pressure jump across the shock in water is so very large. In air, pressure changes across the shock front on the order of 1 to 2 bars are considered large (at least for chemical explosions). In contrast, as pointed out by Richardson, et. al., [Ref. 3], the pressure jump across a shock in water is on the order of kilobars to tens of kilobars. Therefore, the calculations can be simplified by specifying the pressure on the downstream side of the shock front. This is valid since the upstream pressure is so small in comparison to the upstream dynamic pressure $\rho_1 V_1^2/2$ and in comparison to both the downstream pressure and dynamic pressure. The specification of the downstream pressure is accomplished, in the program of this thesis, by allowing the user to input a "pressure multiplication factor (MFACT)", which converts the pressure upstream to a pressure downstream at point 2 which is given by:

$$p_2 = p_1 \text{ X } \text{MFACT X } 1000.0 \qquad (2.24)$$

The factor 1000.0 in equation 2.24 converts the right-hand side of the equation from a pressure in bars to a pressure in kilobars. As an example, if p_1 is 1 bar and MFACT is 5.0, the pressure at point 2 downstream will be 5 kilobars.

It can be shown that the simplifying assumption made above is entirely valid by considering the momentum equation for steady frictionless flow along a streamline. The one-dimensional momentum equation states:

$$p_1 + \rho_1 V_1^2/2 = p_2 + \rho_2 V_2^2/2 \qquad (2.25)$$

Now note that typical values for ρ_1 and V_1 for the problem under consideration are as follows:

$$\rho_1 = 1000 \text{ kg/m} \qquad \text{and} \qquad V = 1500 \text{ m/s}$$

Therefore, $\rho_1 V_1^2/2 = 1.12 \times 10^9$ Pascals or about 11.1 kilobars. Typical values for p_1, the upstream pressure, are on the order of 1 to perhaps a few tens of bars. Thus, it can be safely assumed that p_1 can be ignored compared to the upstream dynamic pressure.

Having specified this pressure, the program of this thesis utilizes the subroutines developed by Fuhs [Ref. 4] to calculate the density and velocity components at a point just downstream of the shock front. Since the geometry of the velocity flow across the shock front is the same in either air or water (see figure 2.4), the velocity components determined in the first step can be used to "jump back across" the shock front to the upstream side where the free-stream Mach number M_1 can be determined.

After determining the conditions on the downstream side of the shock front, the program iterates along the streamlines to determine the cone semi-vertex angle. This iteration is performed using the Busemann graphical construction as for the air case. The major difference between the water and air cases, after the initial shocked conditions have been determined, is that the local speed of sound on a streamline must be determined iteratively, for the water case, by using Fuhs' [Ref. 4] subroutines to determine the density and pressure at each point. Knowing the pressure and density, the speed of sound in the water can be calculated from:

$$c^2 = \left(\frac{nB}{\rho}\right)\left(\frac{\rho}{\bar{\rho}}\right)^n \tag{2.26}$$

which is derived from the modified Tait equation using the definition of the speed of sound as:

$$c^2 = \frac{\partial p}{\partial \rho}\bigg|_S \tag{2.27}$$

These equations, combined with the equations developed in section B of this chapter, constitute all the equations needed to completely solve the supersonic flow over a cone.

TABLE I

Nomenclature

c	Speed of sound
k	Ratio of specific heats (c_p/c_v)
M	Mach number
M_a	Molecular weight of air
p	Pressure
r	Radius in spherical coordinates
R	Radius of curvature of the hodograph streamline
T	Temperature
V	Velocity
V_{max}	Maximum velocity for adiabatic flow
β	Angle the streamline makes with the shock plane (same as the angle σ) in air calculations
δ	Cone semi-vertex angle
θ	Flow direction
ρ	Mass density
σ	Shock angle
ω	Angle in spherical coordinates
Λ	Universal gas constant
$(\ )_1$	Signifies a condition upstream of conical shock or a shock front
$(\ )_2$	Signifies a condition downstream of conical shock or a shock front
$(\ )_s$	Signifies conditions at cone surface
$(\ )_r$	Signifies a component in r-direction
$(\ )_\omega$	Signifies a component in ω-direction

III. DESCRIPTION OF THE COMPUTER PROGRAM

A. PROGRAM LOGIC

As discussed in Chapter 1, the computer program developed for this thesis was written in the FORTRAN programming language. Structured programming practices have been followed throughout in that the program is divided into blocks (or modules) of code, each of which is designed to perform a single calculation sequence. Many of the modules have been placed inline rather than being written as separate subroutines or functions. This was done primarily for ease of use and understanding. In addition, however, most modules were placed inline because the parameter list exchanged between the module and the main program would have been excessively long otherwise. Regardless of whether the code is included inline or as a separate module, the program was written in a manner which ensured that the "side-effects" problem discussed by MacLennan [Ref. 9] did not occur. In addition, the use of FORTRAN's COMMON construct has been studiously avoided to prevent the aliasing problem discussed by MacLennan [Ref. 9]. A module hierarchy chart, which shows the major modules of the program and their interconnections, is given as figure 3.1.

The flow logic of the main computer program is illustrated by figures A.1 through A.7 of Appendix A. These flowcharts show that the initial part of the program (up to line 60) is used to initialize certain key variables and to gather input values from the program user. Note that, as stated in the objectives for this program, all user input is verified to ensure that the values entered fall within prescribed limits. In order to ensure that input values are

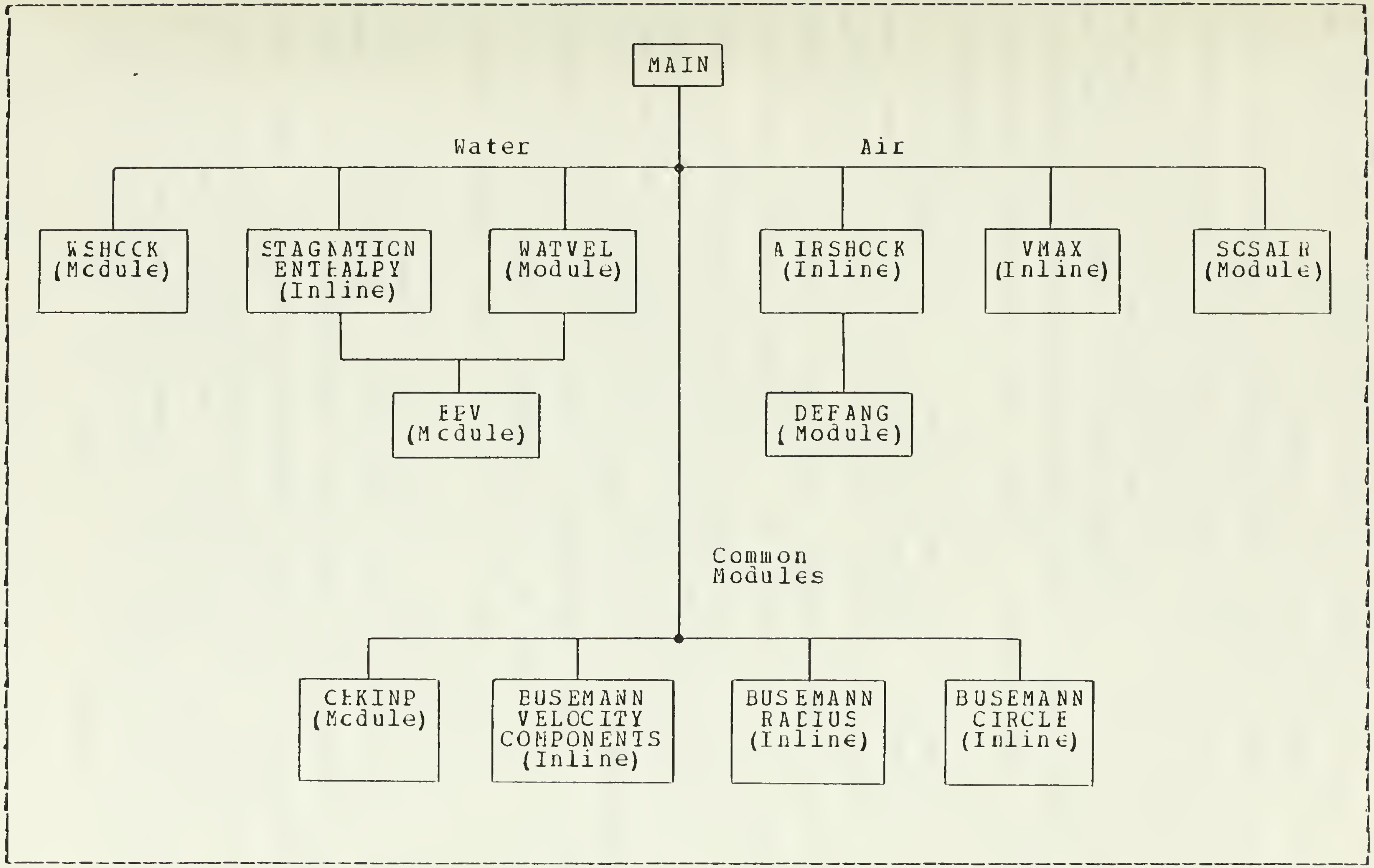

Figure 3.1 Module Hierarchy.

within limits, the subroutine CHKINP is invoked and the input values are passed as parameters. The flow logic of subroutine CHKINP, which is self-explanatory, is included as figures A.8 through A.10 of Appendix A.

After receiving and verifying the user's input, the program begins the calculations required to determine the cone semi-vertex angle. From line 60 to line 400, the program calculates the initial thermodynamic properties of the water or air, as the case may be, and calculates the changes which occur in the fluid properties as a result of the passage of the shock front. As mentioned previously, these calculations for the air case are rather straightforward and, because of this simplicity, the computer code for the air calculation has been placed inline. For the water case, the subroutines WSHOCK and EPV are needed to calculate the required thermodynamic properties. Subroutine WSHOCK calculates the thermodynamic properties of the water at point 1 upstream of the shock front and at point 2, just across the shock front on the downstream or shocked side. Appendix A, figures A.12 and A.13, contains the flowchart which describes the logic of this subroutine. Subroutine EPV was copied, with the author's permission, from Fuhs [Ref. 4] and is fully described in that work. However, the FORTRAN translation of EPV is included in the program listing contained in Appendix C.

Having calculated the initial thermodynamic fluid conditions on both sides of the shock front, the program begins the iteration process required to determine the cone semi-vertex angle. These calculations begin at line 400 of the computer program. This iteration process makes use of the Busemann graphical integration technique discussed by Shapiro [Ref. 8]. First, the radius of the Busemann curve for a point J is calculated using the velocity at that point, the streamline angle at that point, and the value of the angle omega at that point (see figure 3.2). Next, the center of the circle for the Busemann curve at the point J is calculated using the velocity components at point J (which are designated U and V), the radius calculated in the previous step, and the value of the angle omega. This circle center provides the point from which the program "draws" the arc used to calculate the next point on the Busemann curve. Finally, the velocity and the Busemann velocity components, at the next point, J+1, are calculated by "swinging an arc" from the circle center calculated in the previous step. In addition, the streamline angle at the next point is calculated.

The program next tests to see if the cone surface has been reached (line 475). This test is conducted by determining if the absolute value of the difference between the streamline angle and the angle omega is less than a specified test value (which is set to 1 X 10^{-6}). If this difference is less than the test value, the cone surface has been reached. In this case, the program then calculates the thermodynamic fluid conditions at the cone surface and the cone semi-vertex angle and displays the final results of the program. If the difference between the streamline angle and the angle omega is greater than the test value, the cone surface has not been reached. In this case, the program calculates the thermodynamic fluid properties at the next point, J+1, then loops back to line 400 to begin the Busemann graphical integration process for the new point. In the case of a water run, the subroutine WAIVEL is invoked just prior to looping back to line 400. This subroutine calculates the thermodynamic properties of the water at any given point. Appendix A, figures A.14 and A.15, gives a flowchart which demonstrates the logic flow within this subroutine.

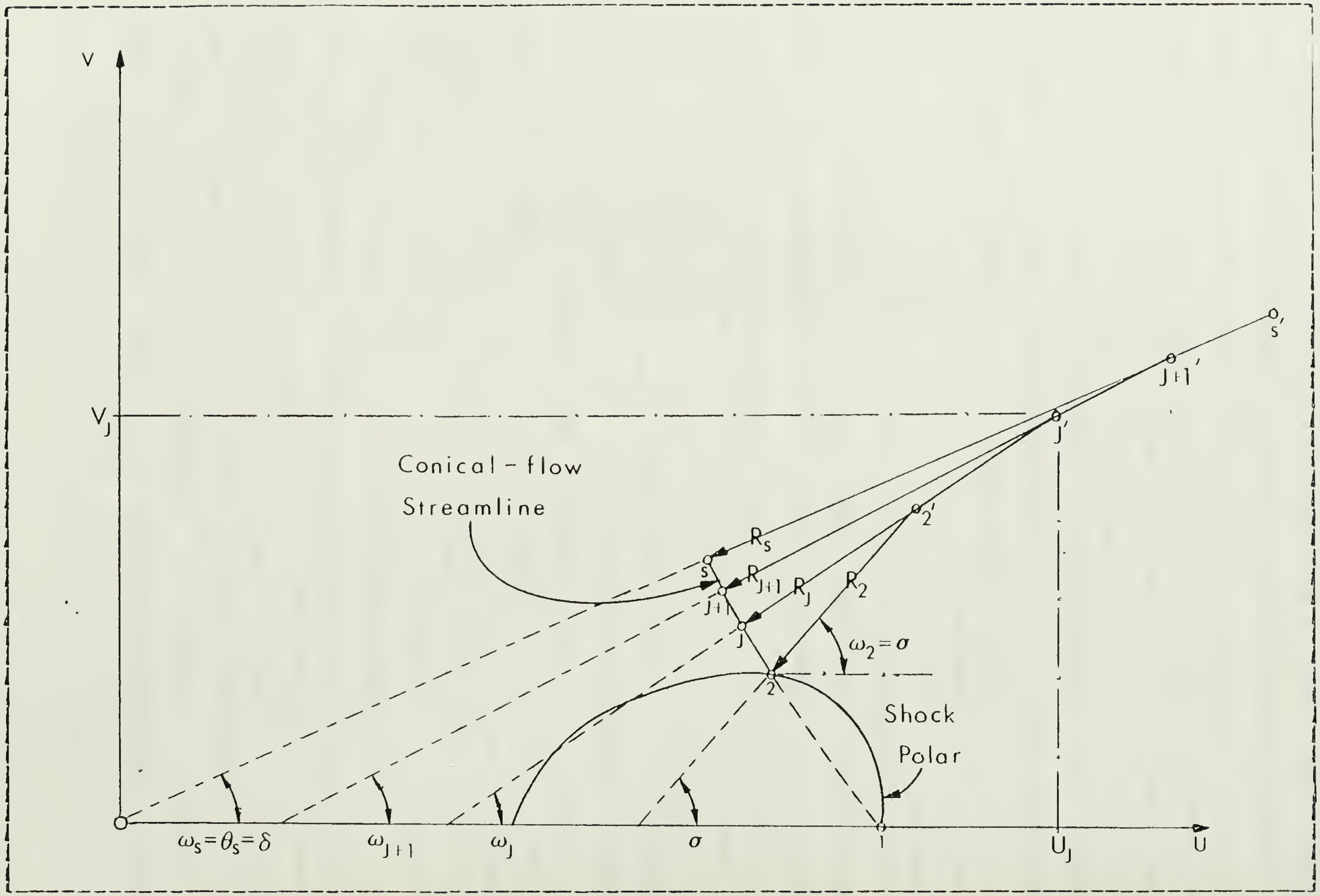

Figure 3.2 Graphical Construction of Cone Flow.

The program includes a feature which allows the user to make repeated executions of the program without the requirement of "reloading" the program between each execution. This feature was included by soliciting a response from the user at line 1500 as to whether he/she wishes to make another execution of the program. An affirmative response causes the program to loop back to statement 1 at the beginning of the Initialization section of the main program. A negative response causes program termination.

Note that certain separately compiled functions included with this program have not been flowcharted because their logic is so straightforward and because the subprograms in question usually consist of only one or two lines of executable code. The functions fitting into this category are: DTOR, RTOD, and SOSAIR.

E. USER INSTRUCTIONS

As discussed in the objectives listed in chapter 1, one goal of this thesis was to ensure that the program presented was easy to use and maintain. Ease of use has been facilitated by including code which verifies that the input provided by the user is within specified ranges. For example, in the water calculations, the pressure at point 2 can be no greater than 100.0 kilobars. This is due to the fact that Fuhs' [Ref. 4] subroutines, which are used to calculate the thermodynamic properties of the water, are based on the work of Richardson, et. al., [Ref. 3], which only gives results up to pressures of 100.0 kilobars. Therefore, while the subroutines could compute the water conditions at pressures greater than 100.0 kilobars, it is uncertain whether the values so calculated would be entirely correct. For this reason, the range of the input values has been restricted. Ease of use is further facilitated by providing clear, meaningful output. Appendix B contains sample outputs generated by the program for the summary and complete print options of the program for both water and air.

One of the major criticisms leveled by computer scientists against the FORTRAN language is for its lack of a requirement for formally defining all variables used within a program. Similarly, FORTRAN has an implicit declaration policy whereby any variable whose first letter is I, J, K, L, M or N is implicitly declared to be of type Integer. In this program, these two unfortunate characteristics of FORTRAN are avoided by requiring explicit declaration of all program variables.

Ease of maintenance and ease of understanding of the program have been achieved through the use of a liberally commented program and through the use of relatively meaningful variable names. In this regard, the FORTRAN language is less than desirable since it limits variable names to six characters. Appendix C contains a fully documented listing of the computer program and all subroutines or functions used by the main program (other than standard library functions such as sin). Tables VI, VII, VIII and IX, located in Appendix A, contain lists of all variable names used in the main program, their meaning, and their MKS (meter-kilogram-second) units, if appropriate. Table X contains a similar variable list for the subroutine WSHOCK and Table XI contains a variable list for the subroutine WATVEL. These two tables are also located in Appendix A.

Finally, this program was developed using the FORTRAN IV language supplied by the IBM Corporation as part of their IBM 370/3033AP computer system which is the main computer system available at the Naval Postgraduate School. As far as is known, no implementation specific features have been included in this program. Therefore, the program should execute on any system which has a FORTRAN compiler. In order to execute the program on the IBM 370 computer system available at the Naval Postgraduate School, the following steps must be accomplished in the order given:

(1) The program must be compiled using the command:

```
FORTHX CONEFLOW
```

This command invokes the FORTRAN H Extended compiler, which is an optimizing, production run compiler supplied as part of the IBM computer system. The program could also be compiled using another FORTRAN compiler such as the FORTGI compiler. Note that the above assumes the program supplied by this thesis has been entered into a file which has the filename CONEFLOW and which is of filetype FORTRAN. Note further that this step need only be performed one time provided errors do not occur.

(2) Next, the libraries of standard subroutines and functions supplied by the computer center must be attached to the file. This is accomplished by issuing the following command:

```
GLOBAL TXTLIB FORTMOD2 MOD2EEH
```

Note that this step need only be performed once provided abnormal terminations do not occur.

(3) In the FORTRAN language, a formatted input/output (I/O) statement requires the definition of an I/O device. For example, at most installations, by default, I/O device 5 is used for input and I/O device 6 is used for output. To utilize the program of this thesis properly, the following commands must be issued to define the I/O devices to be used:

```
FILEDEF 06 DISK CONEFLOW OUTPUT
FILEDEF 07 TERM (PERM)
```

As a result of the above commands, all output from the program will be written into a file on the user's disk named CONEFLOW with a filetype of OUTPUT. It is not strictly necessary to issue the FILEDEF 06 command shown above, but, if not, all output will be written at the terminal and will not be available for printing. The second file definition command (FILEDEF 07) is absolutely required or the program will not operate. This file definition tells the computer that I/O unit 07 is the computer terminal. In the program of this thesis, all input from the user is requested from I/O unit 07. Therefore, if unit 07 is not defined, the program cannot receive any input.

(4) Finally, the program can be executed by issuing the command:

```
LOAD CONEFLOW (START
```

When program execution is completed, a cryptic message of the form R; T=0.01/0.01 16:45:00 will be displayed at the terminal. This is simply a message from the computer's operating system indicating that the task just requested has been completed. The user can now utilize the system's operating commands to review the output from the execution of the program.

If the user desires to execute the program again, return to step (3) of the above procedure. If the output from the previous execution is no longer needed, the same FILEDEF 06 command can be issued. However, if the previous execution's output is needed, the filetype of the FILEDEF 06 command should be changed (e.g. change OUTPUT in the above command to OUTPUT2).

IV. PROGRAM CALCULATION RESULTS

A. PROGRAM RESULTS FOR THE CALCULATION OF CONICAL FLOW IN AIR

The program accurately calculates the flow over a cone in air. Numerous program executions were made for the air case at various upstream Mach numbers and for various shock angles. The results obtained from these program executions are summarized in Tables II and III. The numbers produced by the program were compared for the variables shown in the tables to the tables given in Kinney and Graham [Ref. 5], against the graphs given by Shapiro [Ref. 8], and to the tables produced by Kopal [Ref. 10]. The values given in these sources are included in Tables II and III, where appropriate, along with the calculated results. As can be seen from these two tables, the calculations made by the program give quite accurate results (less than a 1% error in most cases). Certainly, the program is much more accurate than one's ability to read the graphs presented in Shapiro [Ref. 8].

Based on these comparisons, it is believed that the part of the program which calculates supersonic flow over a cone in air is accurate. The importance of this fact is that, by feeling confident that the procedure followed for the air case is valid and that this procedure has been correctly implemented in the programming language, it is safe to assume that the Busemann calculation procedure utilized for the air case can be accurately applied to the water case provided the water conditions at each point are calculated correctly.

TABLE II

Program Results and Comparisons

Shock Angle	Cone Angle		Mach Number at Cone Surface		Drag Coefficient	
(1)	Prog.	(1)	Prog.	(2)	Prog.	(2)
19.719	5.7098	5.0	2.86	2.9	0.036	0.04
20.458	7.6728	7.5	-	-	-	-
21.716	10.042	10.0	2.71	2.75	0.089	0.08
23.360	12.5096	12.5	-	-	-	-
25.214	15.0107	15.0	2.50	2.5	0.174	0.18
27.375	17.5024	17.5	-	-	-	-
29.635	20.0084	20.0	2.29	2.3	0.285	0.29
32.018	22.5080	22.5	-	-	-	-
34.514	25.0091	25.0	2.06	2.05	0.420	0.44
39.822	30.0183	30.0	1.83	1.85	0.576	0.575
45.555	35.0276	35.0	1.59	1.65	0.751	0.75
51.860	40.0361	40.0	1.35	1.38	0.941	0.94
59.263	45.0463	45.0	1.08	1.2	-	-

Notes:

(1) The shock angle values were taken directly from Kopal. In Kopal's work, the entering parameters are the cone semi-vertex angle and upstream Mach number. This is the reverse of the way the program operates. However, using Kopal's values allows for comparisons to be made more easily.

(2) The numbers in these columns were read from the graphs presented in Shapiro. Thus, the numbers are subject to interpretation depending upon how well one reads graphs. Regardless, the numbers presented were read from the graphs at the whole number cone angles given in the same row (e.g. the first row of the table was read at the cone angle of 5.0 degrees). Thus, there is some room for error here since the program values are calculated at slightly different cone angles. As one can see, however, the values match relatively well.

(3) The following variables were held constant at the values given for all program executions: T_1 = 298.16 Kelvin; P_1 = 101300.0 Pascals; and M_1 = 3.0.

TABLE III

Program Results and Comparisons (cont'd.)

Shock Angle	Pressure Ratio (P2/P1)		Sound Velocity Ratio (C2/C1)		Temperature Ratio (T2/T1)	
(1)	Prog.	(2)	Prog.	(2)	Prog.	(2)
19.719	1.0287	1.0283	1.0040	1.0039	1.0081	1.0083
20.458	1.1161	1.1163	1.0158	1.0157	1.0319	1.0321
21.716	1.2708	1.2710	1.0350	1.0350	1.0713	1.0710
23.360	1.4841	1.4838	1.0590	1.0589	1.1214	1.1217
25.274	1.7473	1.7475	1.0857	1.0853	1.1788	1.1785
27.375	2.0533	2.0532	1.1142	1.1138	1.2415	1.2416
29.635	2.4006	2.4006	1.1442	1.1440	1.3092	1.3094
32.018	2.7849	2.7851	1.1754	1.1752	1.3815	1.3814
34.514	3.2043	3.2043	1.2076	1.2079	1.4582	1.4579
39.822	4.1396	4.1395	1.2746	1.2744	1.6245	1.6249
45.555	5.1850	5.1888	1.3439	1.3425	1.8061	1.8076
51.860	6.3285	6.3314	1.4149	1.4179	2.0020	2.0036
59.263	7.5905	7.5935	1.4888	1.4914	2.2164	2.2186

Notes:

(1) The shock angle values were taken directly from Kopal. In Kopal's work, the entering parameters are the cone semi-vertex angle and upstream Mach number. This is the reverse of the way the program operates. However, using Kopal's values allows for comparisons to be made more easily.

(2) The numbers in these columns were read from Table IX of Kinney and Graham. A linear interpolation between the values listed in Table IX of Kinney and Graham was made to arrive at the numbers presented in these columns. This may account for some of the variation seen between the values calculated by the program and those of the reference material. As can be seen from the table, even these variations are quite small.

(3) The following variables were held constant at the values given for all program executions: T_1 = 298.16 Kelvin; P_1 = 101300.0 Pascals; and M_1 = 3.0.

E. PROGRAM RESULTS FOR THE CALCULATION OF CONICAL FLOW IN WATER

As mentioned in the previous section, the calculation procedure was verified for the air case by comparison with known results. This comparison showed the method used in the calculations was correct. The next step taken was to determine if the subroutines used to calculate the thermodynamic properties of the water gave results comparable to those of Richardson, et. al, [Ref. 3], and Fuhs [Ref. 4]. Calculations were performed which gave results which could be compared to Table II of Fuhs [Ref. 4]. Table IV presents a comparison between the numeric results calculated by the program and those given by Fuhs. As can be seen, these results are, within reasonable error, remarkably similar. This is not unexpected since the subroutines used by the main program to calculate these variables are essentially direct language translations of Fuhs' programs. Therefore, the results should be similar.

Having verified that the procedure utilized in the calculation of the cone semi-vertex angle was correct (through the air results) and having verified that the program correctly calculated the thermodynamic properties of the water at any point, it was believed that the program could be executed for the water case, for various initial conditions, with certainty that the results so calculated would be accurate. However, a question arose as to whether the calculations performed in the water case were independent of the upstream thermodynamic properties of the water. In air, the only quantities needed to calculate the cone semi-vertex angle are the upstream Mach number (M_1) and the shock angle (σ) (e.g. see figure 17.7(a) of Shapiro [Ref. 8]). Thus, for air, the calculation of the cone semi-vertex angle is independent of the upstream temperature or pressure, and, therefore, universal curves can be drawn

TABLE IV

Properties of Sea Water at a Shock Front

P kbar	Water Velocity at Point 2 (m/s)			Shock Velocity (m/s)			Speed of Sound at Point 2 (m/s)			Specific Volume at Point 2 (cm3/gm)		
	Prog	Fuhs	RAH	Prog	Fuhs	RAH	Prog	Fuhs	RAH	Prog	Fuhs	RAH
5	256	258	257	1934	1939	1930	2193	2201	2190	.8581	.8572	.8593
10	430	434	433	2298	2306	2290	2724	2736	2720	.8036	.8026	.8040
20	693	698	698	2856	2868	2845	3515	3533	3510	.7490	.7480	.7483
30	899	907	905	3299	3315	3285	4130	4152	4125	.7194	.7185	.7186
40	1076	1084	1080	3677	3695	3665	4646	4671	4640	.6995	.6987	.6989
50	1233	1243	1240	4010	4031	4000	5097	5125	5095	.6848	.6840	.6842
60	1376	1386	1385	4313	4336	4300	5503	5533	5495	.6735	.6727	.6728
70	1507	1518	1515	4595	4620	4585	5876	5909	5870	.6646	.6639	.6641
80	1626	1638	1635	4864	4892	4855	6229	6265	6225	.6582	.6576	.6579
90	1735	1747	1740	5128	5159	5120	6569	6609	6570	.6542	.6539	.6542

Notes: (1) The conditions upstream of the shock front were established to be as follows: temperature = 0° C; salinity = 0.7 m NaCl; acoustic velocity, c_o, = 1443 m/s.

(2) The values for the two columns on the right of each major division identified as Fuhs and RAH, were taken from Table II of Fuhs [Ref. 4].

which describe the cone flow under all conditions as is shown by Shapiro [Ref. 8]. Whether universal results for the water case could be obtained was not so clear; this point is now discussed.

Liepmann and Roshko [Ref. 11] discuss the general thermodynamics of fluids and introduce the concept of a "thermal" equation of state. In general, the thermal equation of state for any fluid is given by:

$$f(p, \rho, T) = 0 \tag{4.1}$$

For a perfect gas, the thermal equation of state is given by:

$$pv = RT \tag{4.2}$$

where $R = \Lambda / M_a$. Another equation which relates the thermodynamic variables ϵ, v, and T is the "caloric" equation of state, which, in a general form, is given by:

$$f(\epsilon, v, T) = 0 \tag{4.3}$$

A calorically perfect gas is defined by:

$$e = c_v T \tag{4.4}$$

or by:

$$h = c_p T \tag{4.5}$$

where ϵ is internal energy and h is enthalpy. Thus, the heat capacity (either c_v or c_p) is a constant for a calorically perfect gas.

Liepmann and Roshko [Ref. 11] demonstrate that the thermal and caloric equations of state are thermodynamically related by:

$$\frac{\partial h}{\partial p}\bigg|_T = v - T\frac{\partial v}{\partial T}\bigg|_p \tag{4.6}$$

Introducing equation 4.2 into equation 4.6 results in the right-hand side of equation 4.6 becoming equal to zero. Thus, for a thermally perfect gas, the enthalpy is independent of the pressure, and, hence, enthalpy is a function only of temperature. Therefore, a necessary condition for equation 4.5 to be valid is to have a thermally perfect gas.

For water, Richardson, et. al., [Ref. 3], demonstrate that the heat capacities are functions of the temperature of the water. Thus, equations 4.4 and 4.5 are not valid for water, and water is not a calorically perfect fluid.

If a fluid is both thermally and calorically perfect, the conditions across a normal shock front depend only on $k = c_p/c_v$ and on the freestream Mach number. Conversely, if the fluid is not thermally and calorically perfect, additional variables must be specified in order to define the shock conditions. Extending the argument, one can state that the supersonic flow of a thermally and calorically perfect fluid over a cone is a function only of the heat capacity ratio, k; the freestream Mach number, M_1; and the shock front angle, σ. As a result, a single graph is sufficient to represent all flows over the cone. This is the condition for a perfect gas.

In contrast, when the fluid is not thermally and calorically perfect, the solution for supersonic conical flow is dependent upon variables other than k, M_1, and σ. Consequently, a universal graph of the shock angle versus the freestream Mach number, with the cone angle as an entry parameter, does not exist. Since water is not a thermally and calorically perfect fluid, it is not possible to plot universal graphs of the variables of interest.

To gain insight into the sensitivity of the supersonic conical flow in water to the upstream conditions, a study was conducted in which calculations were performed to determine whether the cone semi-vertex angle is relatively independent of the upstream temperature or whether various program executions are required to show the variation of cone semi-vertex angle with water temperature.

The determination discussed above was made by holding the upstream pressure, the shock angle, and the upstream Mach number constant. Then the upstream temperature was varied to determine the variation in the cone semi-vertex angle. In order to hold M_1 constant, the pressure multiplication factor, described in Chapter II.D, was varied until the Mach number upstream for the new temperature matched (within reasonable accuracy) the upstream Mach number for the original temperature. Note that "reasonable accuracy" for matching of the upstream Mach numbers meant matching the numbers to the fourth decimal place. Table V provides the results from these calculations. As can be seen from this table, the cone semi-vertex angles calculated for the various temperatures are all of approximately the same value (within 1%). The pressure downstream of the shock wave varies by 13% as can be seen in Table V. While there is small variation between the values for the cone semi-vertex angles, it must be remembered that, for practical purposes, these variations are very small and can be safely ignored in examining the flow over a cone in water. Thus, it was concluded that the calculation of the cone semi-vertex angle in water can be conducted independently of the water temperature (at least for the accuracy required for the calculation of the movement of the metal jet from a shaped-charge).

TABLE V

Cone Semi-vertex Angle Variation with Temperature

Upstream Temperature (Kelvin)	Upstream Mach Number	Pressure Multiplication Factor	Cone Semi-Vertex Angle (Degrees)
273.16	3.197999	1.000	4.9866
283.16	3.198033	1.051	4.8844
293.16	3.198458	1.090	4.8199
303.16	3.198541	1.115	4.7786
313.16	3.198482	1.128	4.7565
323.16	3.198257	1.130	4.7513
333.16	3.198505	1.125	4.7653
343.16	3.198392	1.111	4.7923
353.16	3.198447	1.090	4.8322
363.16	3.198063	1.062	4.8786
373.16	3.198013	1.031	4.9321

Note: In the execution of the program, the upstream pressure was held constant at 101300.0 Pascals and the shock angle was held constant at 20 degrees.

This conclusion is true provided the upstream Mach number is held constant by varying the pressure behind the shock front through the mechanism of the pressure multiplication factor. Having determined that the calculations can be conducted independently of the temperature, plots of the variables of interest in water were made in a manner similar to the graphs for air shown in Shapiro [Ref. 8].

Finally, having determined that the calculations could be made independently of the upstream temperature and that these calculations would be accurate (within 1% for cone angle and 13% for pressure), it was decided to plot various parameters of interest for the flow over a cone in water. These graphs are presented as follows:

(1) Figure 4.1 is a plot of the freestream Mach Number versus the shock angle with the cone semi-vertex angle as an entry parameter. This graph is the water analog to Figure 17.7.(a) of Shapiro [Ref. 8].

(2) Figure 4.2 is a plot of the freestream Mach number versus the Mach number at the cone surface with the cone semi-vertex angle as an entry parameter. This graph is the water analog to Figure 17.7.(c) of Shapiro [Ref. 8].

(3) Finally, Figure 4.3 is a plot of the freestream Mach number versus the drag coefficient (c) with the cone semi-vertex angle as an entry parameter. This graph is the water analog to Figure 17.7.(f) of Shapiro [Ref. 8].

It should be noted here that the graphs developed and presented in this thesis were obtained through repeated executions of the CONEFLOW program, which generated approximately 3600 data points per graph. These data points were pre-processed to remove entries with a Mach number of greater than 5.0 (a convenience for ease of presentation only). The pre-processed data points were passed to an interpolation and smoothing program developed by Franke [Ref. 12] which used a quadratic Shepard's method to smooth the data points generated. After smoothing, the data was post-processed to remove erroneous interpolation points which were generated because the original data was irregular. Finally, the processed data was passed to a FORTRAN program developed by the author of this thesis which utilized the Contouring feature of the DISSPLA graphing routines of the ISSCO company. The results of this extensive processing are the graphs presented in this thesis. Because of the irregularity of the original data (which causes difficulties in the quadratic Shepard's interpolation method), and because of the smoothing required by the contouring routines of DISSPLA, the graphs presented are somewhat rough and should, therefore, only be used for "back of the envelope" calculations. Accurate results are provided by the CONEFLOW program which should be utilized for more precise work.

Finally, one of the primary reasons for translating the original Basic programs into a higher-order language (in addition to the desire to make the programs more accessible) was to speed the execution of the program. In this regard, the work of this thesis has more than accomplished this result. It must be mentioned here that the program will execute extremely fast, considering all the iterations needed, provided the computer system in use is not heavily in use at the time. For example, at the Naval Postgraduate School, when the computer system is relatively free, the program operates so quickly that one execution requires less than 5 seconds (from the time of the last user input to the time of the request by the program for an indication of whether another execution is desired). Conversely, when the computer system is being utilized at near its capacity, the same response time can increase up to about 2 to 3 minutes. This result, however, is symptomatic of any computer system being utilized at near capacity and is not a result or a reflection of the program's structure or design.

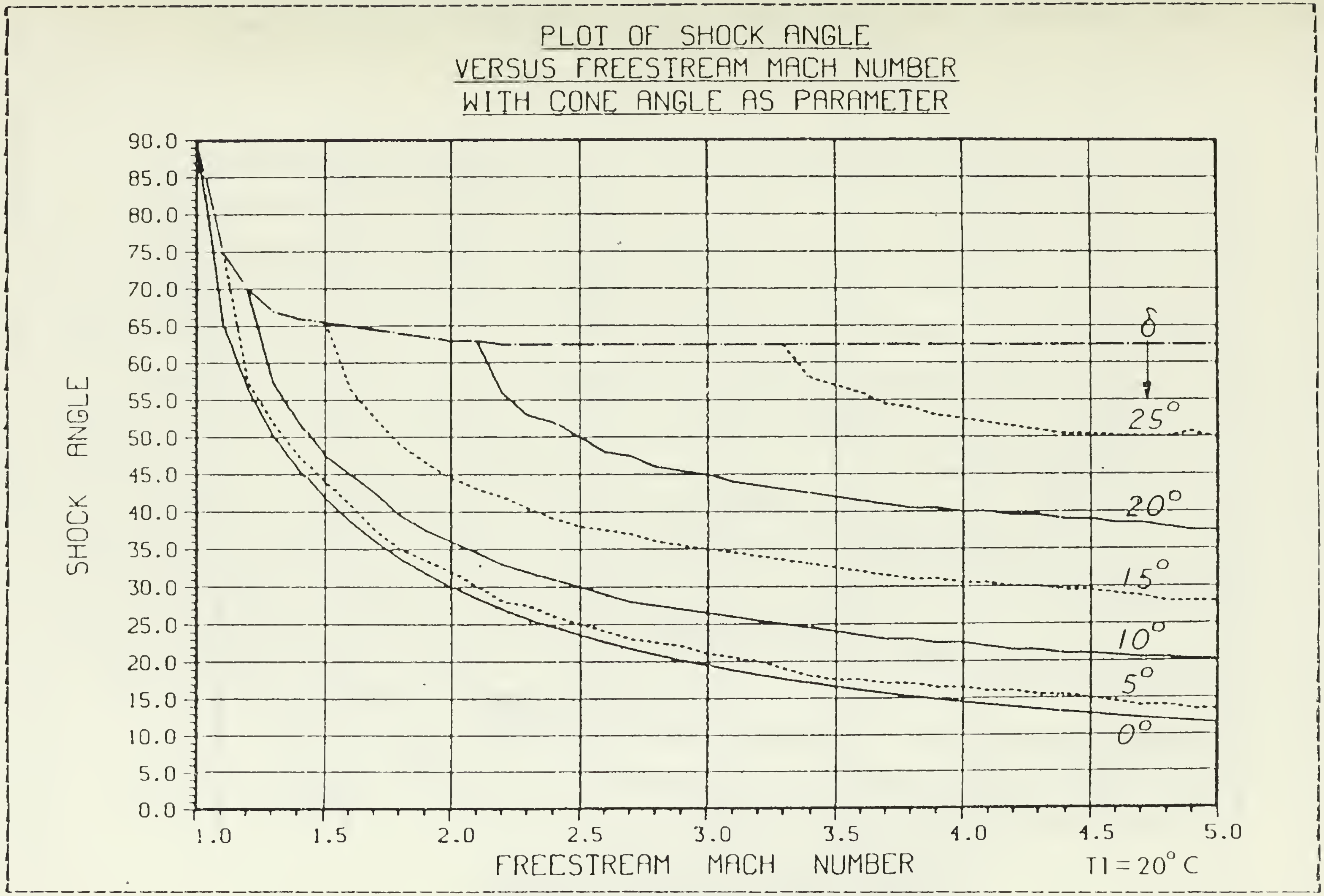

Figure 4.1 Freestream Mach Number vs Shock Angle for Supersonic Flow in Water.

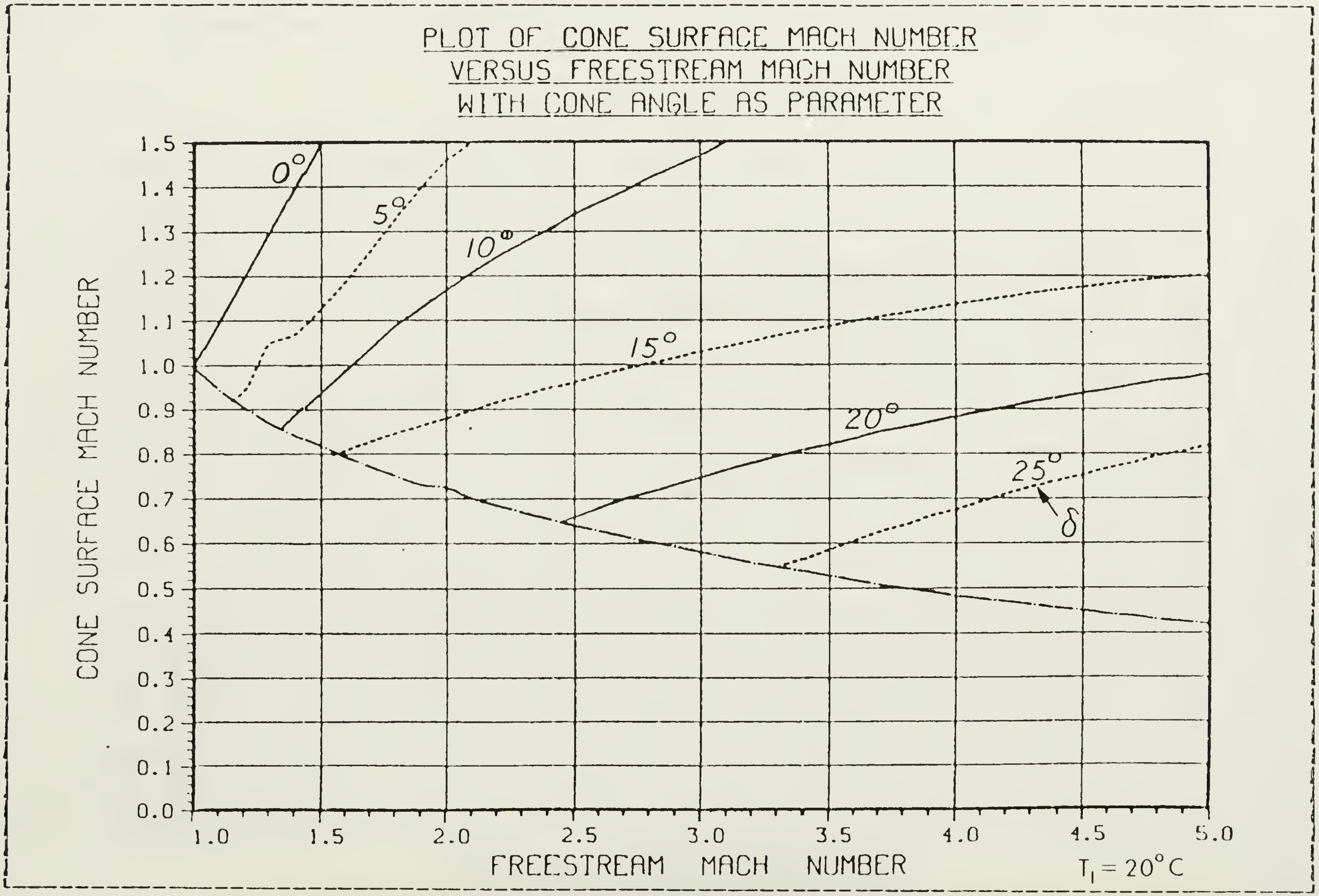

Figure 4.2 Freestream Mach Number vs Surface Mach Number for Supersonic Flow in Water.

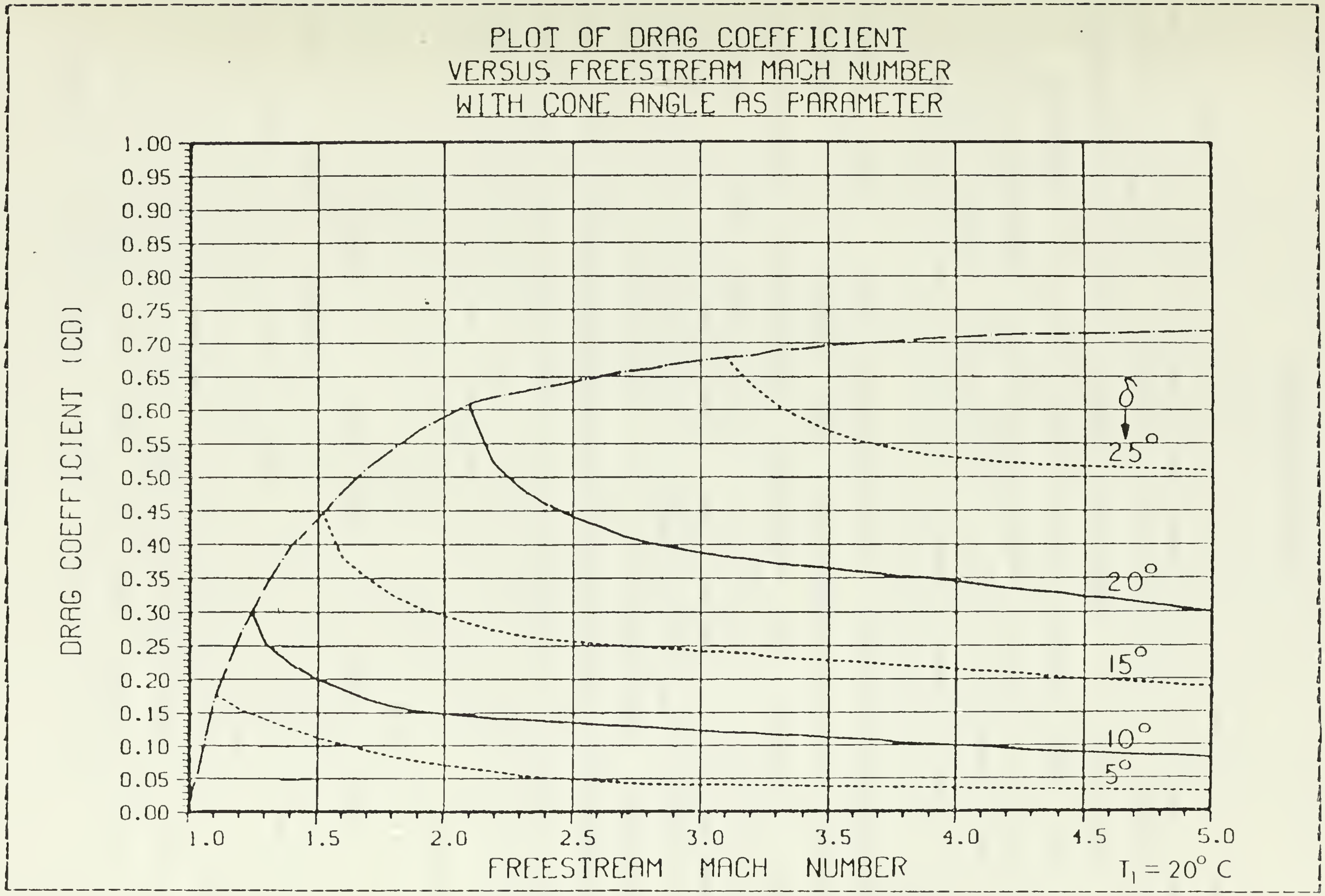

Figure 4.3 Freestream Mach Number vs Drag Coefficient for Supersonic Flow in Water.

V. CONCLUSIONS

As discussed in the Introduction, this thesis had certain goals which it was desired to achieve. It is believed that these goals have all been successfully met. The program of this thesis has been successfully translated from Basic into FORTRAN as desired. In the translation of the original program, modern structured programming practices have been followed to the greatest extent possible. Finally, the program presented is quite "user-friendly" and is very well-documented.

As was demonstrated in the results chapter, this program calculates correct results for each of the air or water cases. It is mentioned here that, as pointed out frequently by computer scientists, it is virtually impossible to test a program for all possible cases. Therefore, no program is entirely error-free. It is believed that the program of this thesis is as free of errors as is possible without exhaustive, time-consuming and extremely expensive testing.

As a result of having met the three goals mentioned in the first paragraph, it is believed that the program of this thesis will provide an excellent working tool for researchers in the field of shaped-charge jet penetration in water. The fact that the program is well-documented will make any modifications or extensions to the program, should that be required or desired, easy to perform. Further, since the program is easy to use, only a cursory knowledge of computers is needed in order to utilize the program. These features are requirements for any computer program which is to be used by scientists as a tool. Too often, computers and their programs require that the people who simply desire to use their capabilities must learn a great deal of detail about the computer in order to use the tool. In this program, an understanding of FORTRAN is not required in order to utilize the program or its design. Unfortunately, due to the operating system of the IBM 370 computer system, the same cannot be said about the steps required in order to actually use the program. Some familiarity with the computer system in use at the users location will be a necessity in order to operate the program correctly.

Finally, as mentioned in the Introduction to this thesis, the program will provide an excellent test case and comparison model for a computer program which models the actual flow of the metal jet from an explosive shaped-charge fired through the water. The actual situation is a blunt-nosed, rather than a sharp-pointed conical, flow problem which is much more difficult to solve. Therefore, a known solution and methodology of solution for the easier problem is a necessary first step to the solution of the larger problem. It is believed that the program of this thesis will serve as this necessary first step. Further, it is believed that, when the equations solving the actual flow problem are develcped in their final form, the program of this thesis, due to its ease of modification, will serve as the programming model for the program which calculates the blunt-nosed flow problem.

APPENDIX A

PROGRAM FLOWCHARTS

This appendix contains the logic flowcharts of the main program and its subroutines (i.e. those which have not been described elsewhere or which are not part of standard computer center libraries). The language is kept rather general so the overall logic of the program can be demonstrated. If the reader desires to know how a particular logic sequence is implemented, he need only refer directly to the program segment which the logical flowchart is describing. The flowcharts included in this appendix are listed below:

(1) The Main Program Flowchart consists of Figures A.1 through A.7.

(2) Function CHKINP consists of Figures A.8 through A.10.

(3) Subroutine DEFANG consists of Figure A.11.

(4) Subroutine WSHOCK consists of Figures A.12 and A.13.

(5) Subroutine WATVEL consists of Figures A.14 and A.15.

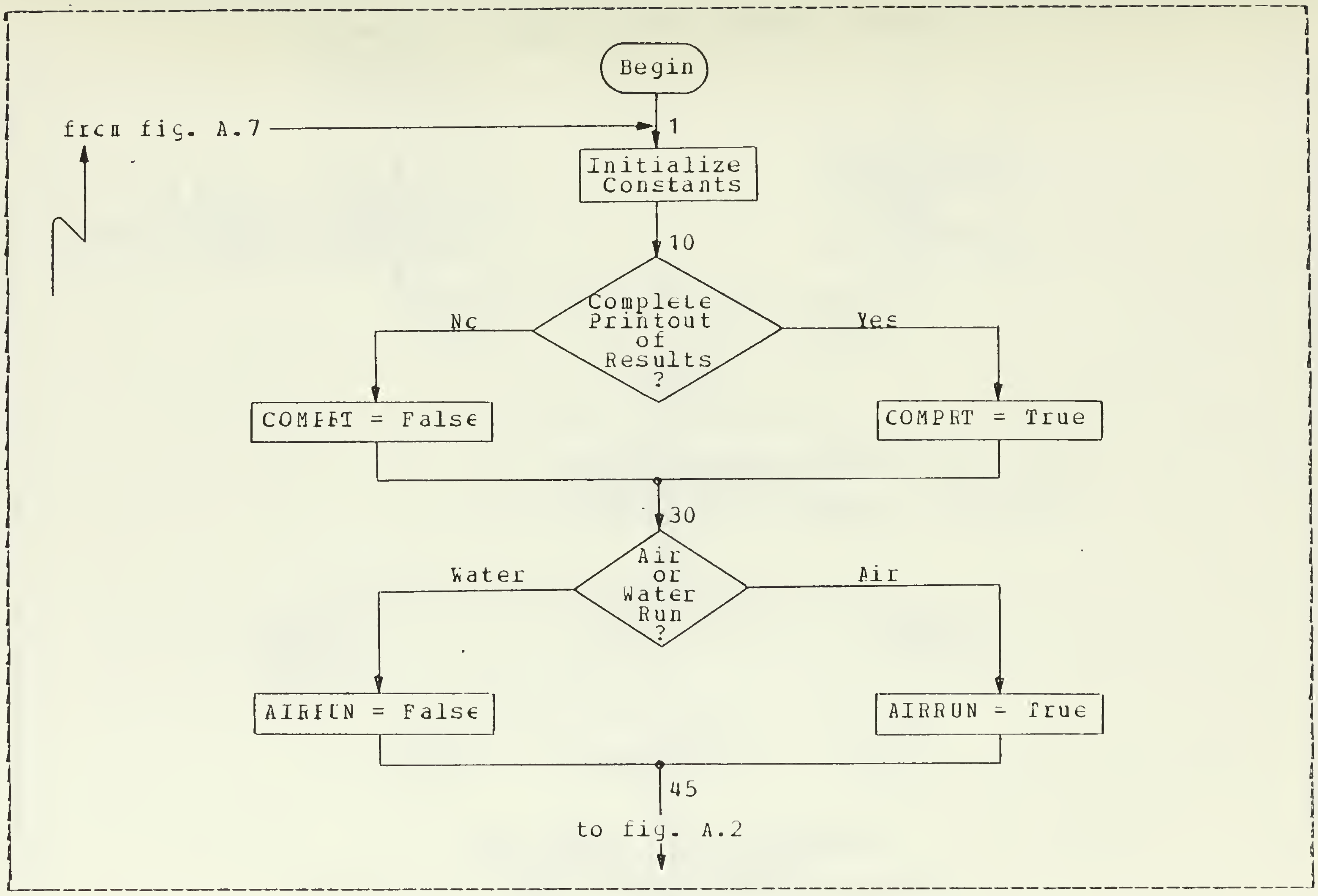

Figure A.1 Main Program Flowchart.

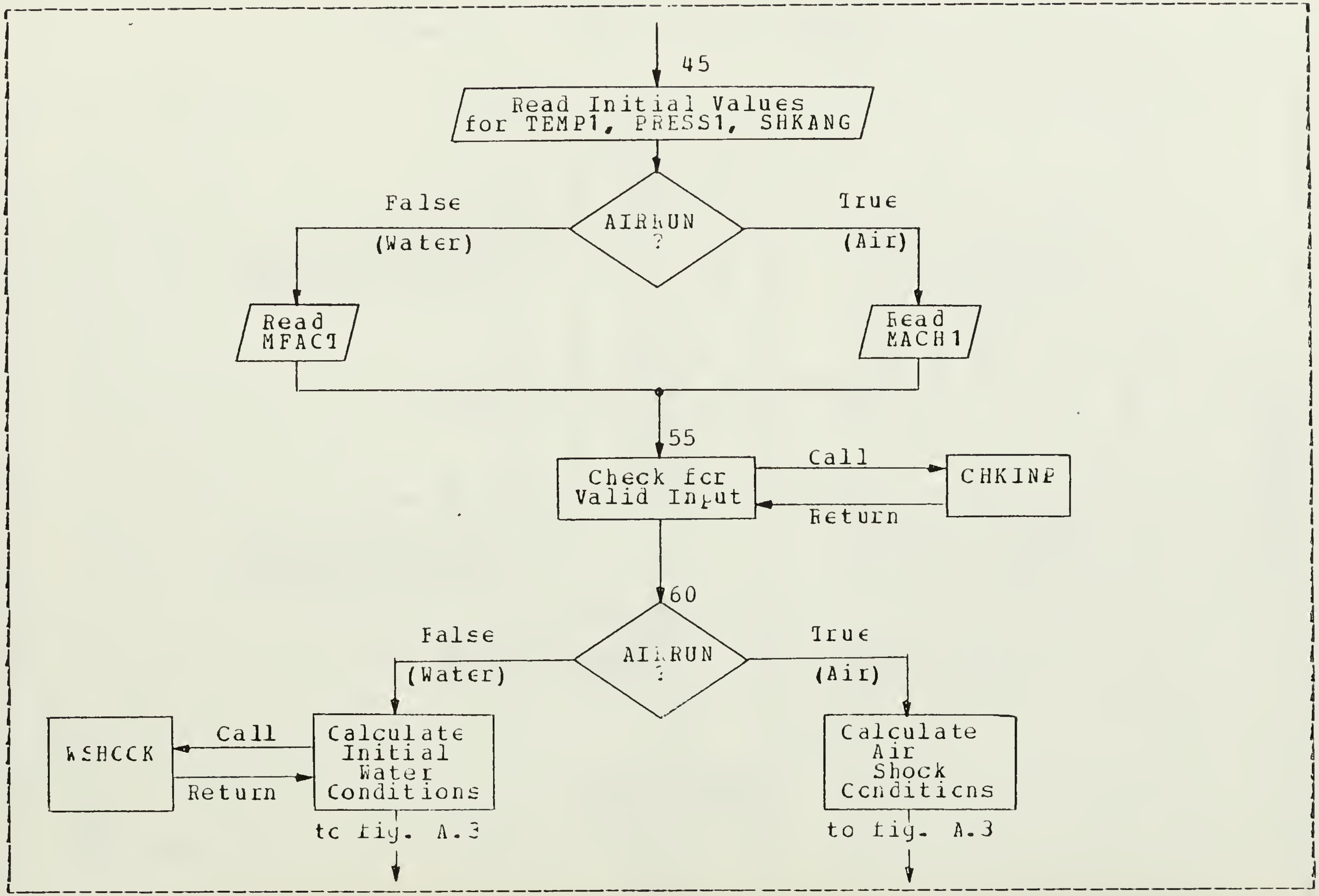

Figure A.2 Main Program Flowchart (cont'd.).

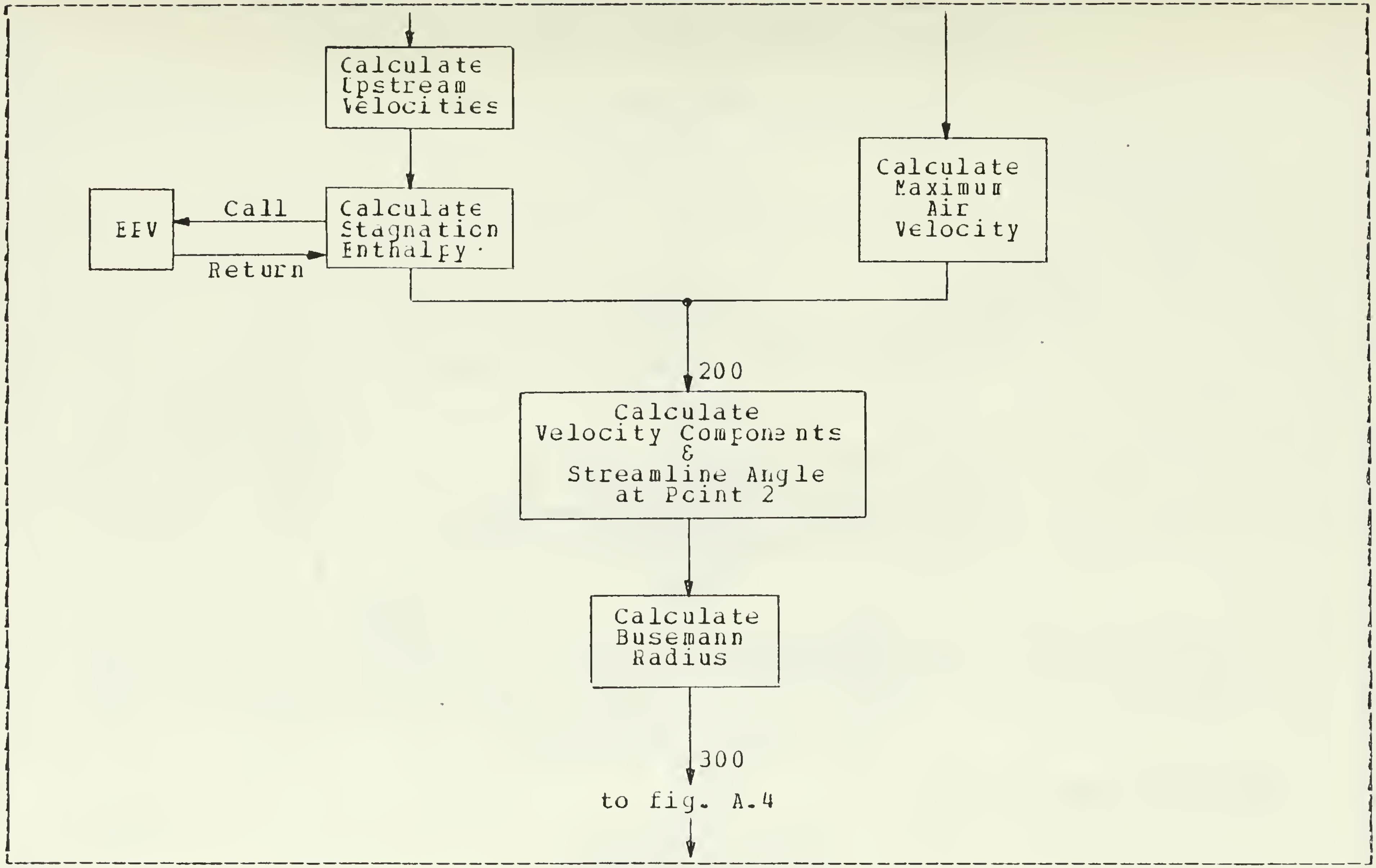
Calculate Upstream Velocities
Calculate Stagnation Enthalpy
Call
Return
EEV
Calculate Maximum Air Velocity
200
Calculate Velocity Components & Streamline Angle at Point 2
Calculate Busemann Radius
300
to fig. A.4

Figure A.3 Main Program Flowchart (cont'd.).

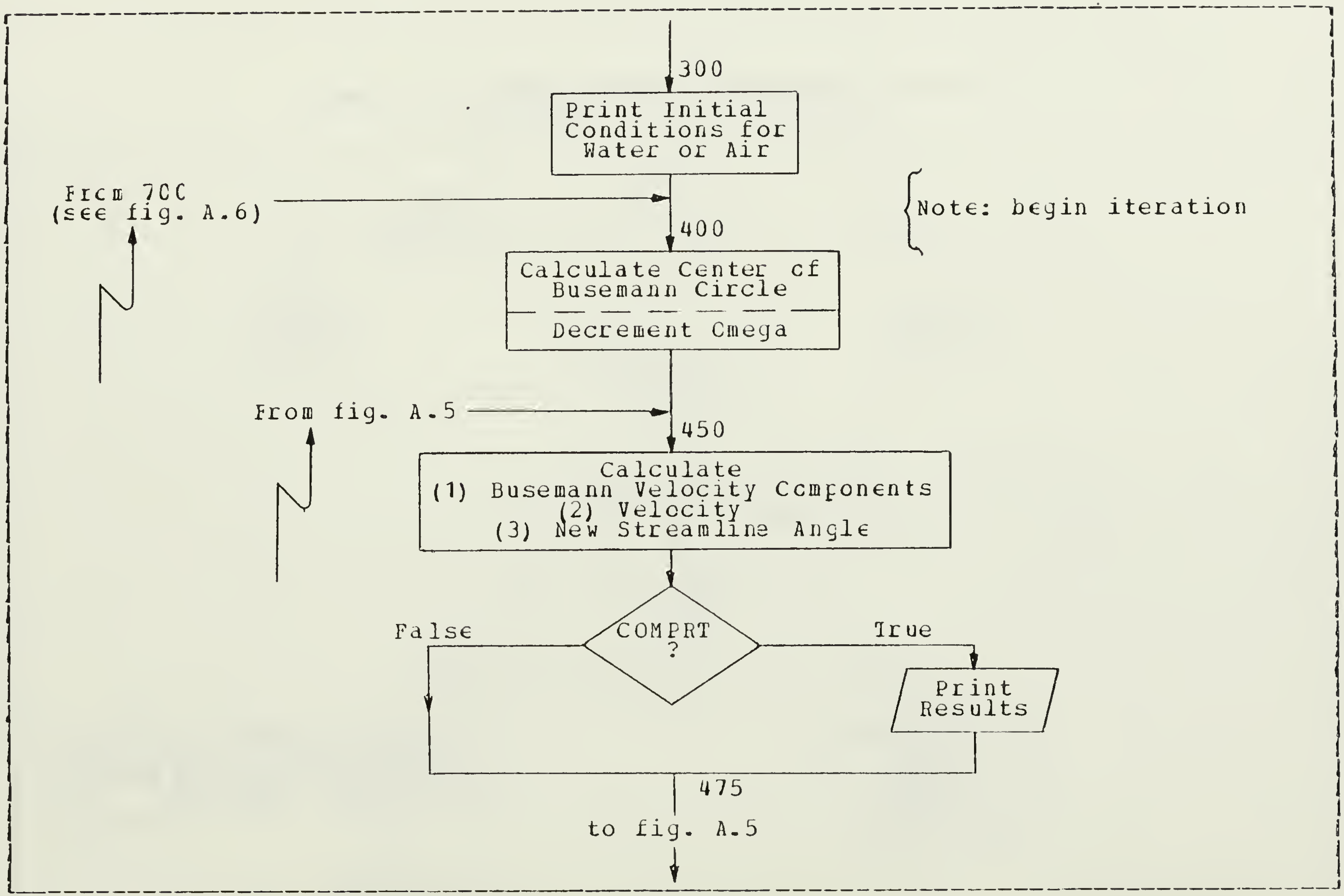

Figure A.4 Main Program Flowchart (cont'd.).

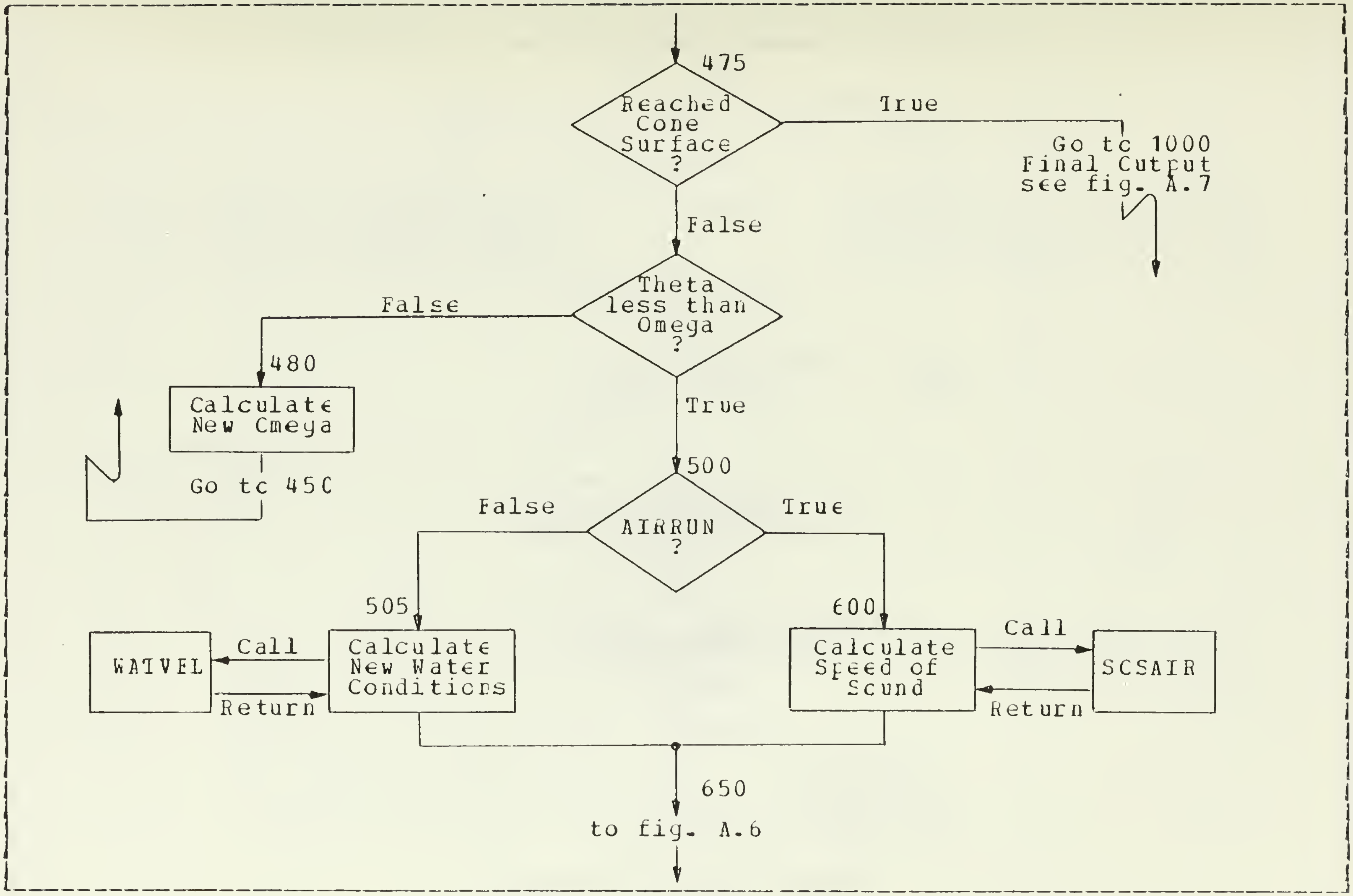

Figure A.5 Main Program Flowchart (cont'd.).

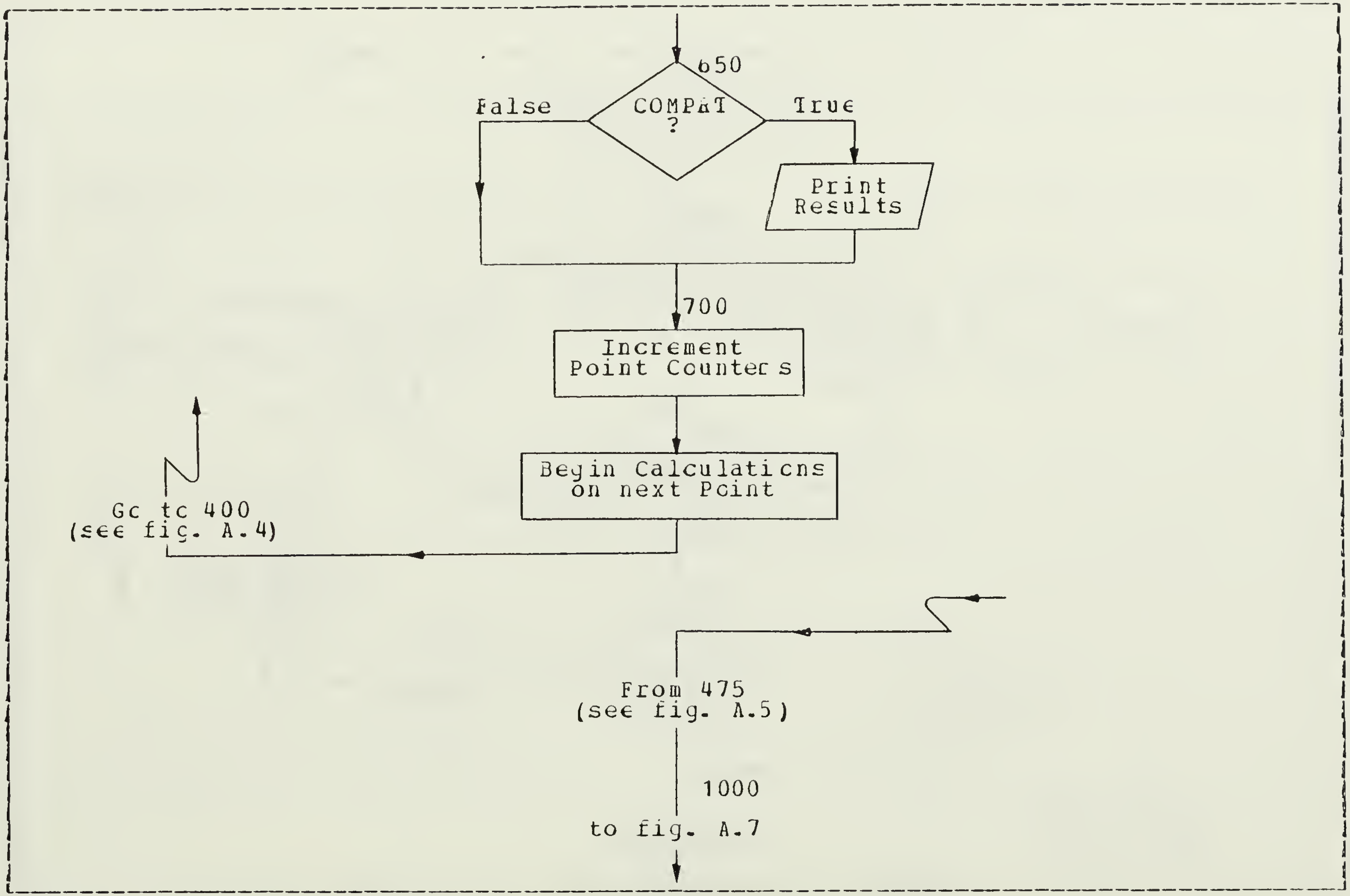

Figure A.6 Main Program Flowchart (cont'd.).

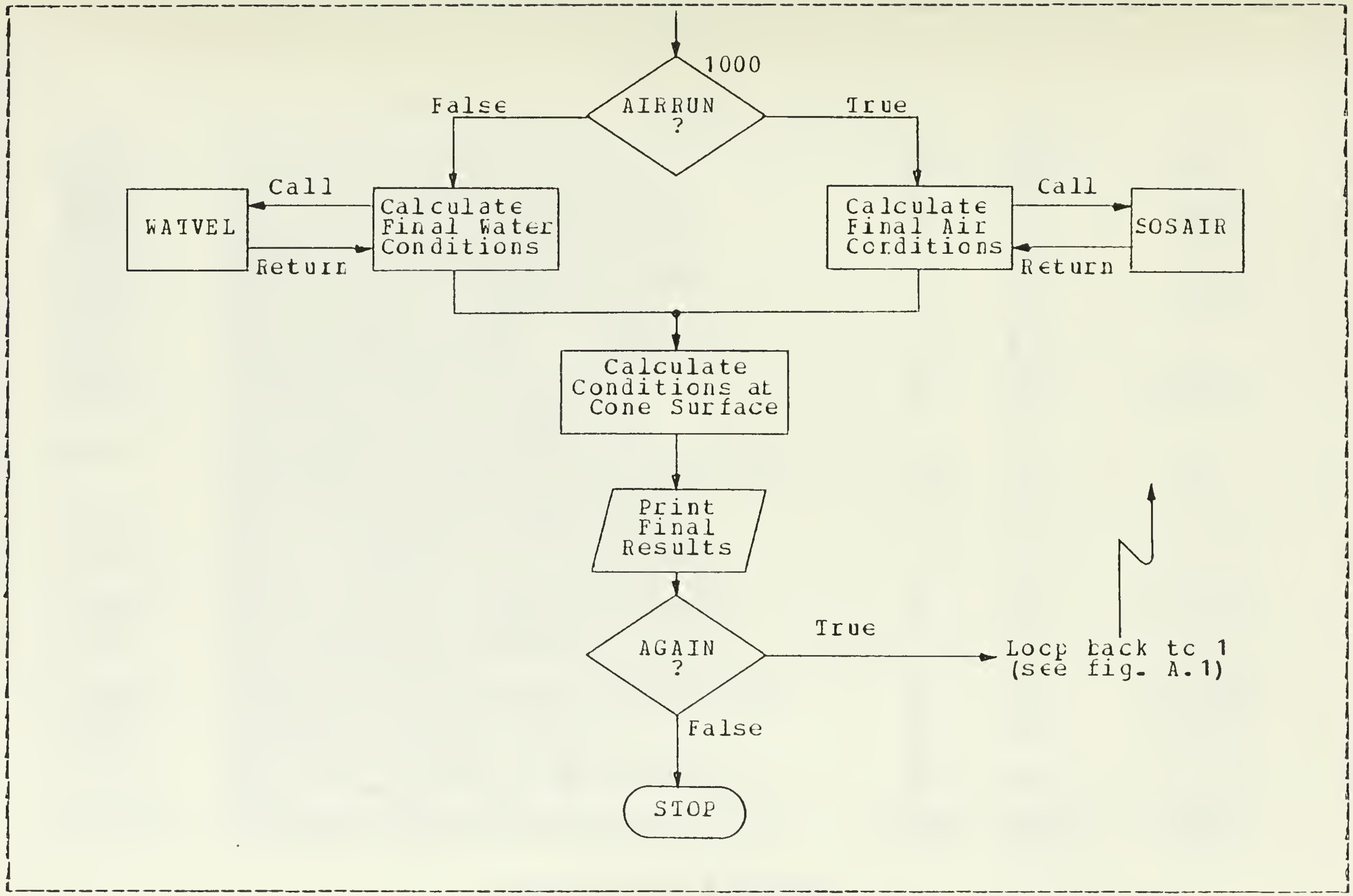
1000
AIRRUN ?
False
True
Call
WATVEL
Return
Calculate Final Water Conditions
Calculate Final Air Conditions
Call
SOSAIR
Return
Calculate Conditions at Cone Surface
Print Final Results
AGAIN ?
True
Loop back to 1 (see fig. A.1)
False
STOP

Figure A.7 Main Program Flowchart (cont'd.).

TABLE VI

Main Program Variables

Variable	Variable's Meaning or Use	Type	Module	Units
AGAIN	Used to determine if another run of the program is desired	C	F.C.	-
AIRM	Molecular Weight of Air	R	A.S.	-
AIRR	Universal Gas Constant	R	A.S.	J/mol-K
AIRRUN	Indicates if run is for air or water	I	M.	-
ANGDEF	Deflection Angle after crossing the shock front	R	A.S.	Radians
ARESP	Test value to determine if run is for air or water	C	I.	-
ARGUM1	Holds argument from an air shock formula passed to SQRT function	R	A.S.	-
ARGUM2	Holds argument from an air shock formula passed to SQRT function	R	A.S.	-
AWRESP	User response to type of run option - an Input variable	C	I.	-
C	An Array which holds the speed of sound at each point J	DP	M.	m/s
COMPRT	Indicates if a complete or summary print is desired	I	M.	-
CONANG	Cone Semi-Vertex Angle	R	F.C.	Degrees
CREF	Reference Speed of Sound on Upstream side of shock front	DP	W.S. W.V.	m/s
DEN	Value of denominator in formula for Busemann Radius	DP	B.R.	-
DENSCS	Density at the Cone Surface	R	F.C.	kg/m3
DENSTY	Array which holds value of the Density at each point J	DP	M.	kg/m3
DENSY1	Density at Point 1 upstream	DP	M.	kg/m3
DENSY2	Density at Point 2 downstream	DP	M.	kg/m3
DRAGCO	Drag Coefficient for the Cone	R	F.C.	-
ENERG	Energy term returned by EPV	DP	S.E.	J/kg
ENTH2	Enthalpy at Point 2 downstream	DP	S.E.	J/kg
ENTHCS	Enthalpy on cone surface - water case	R	F.C.	J/kg
ENTHIP	Array holding values of Enthalpy at each point J	DP	W.V.	J/kg

TABLE VII

Main Program Variables (cont'd.)

Variable	Variable's Meaning or Use	Type	Module	Units
GAMMA	Ratio of Specific Heats (c /c)	R	A.S.	-
GAMMA1	Value of an expression of Gamma	R	A.S.	-
GAMMA2	Value of an expression of Gamma	R	A.S.	-
GAMMA3	Value of an expression of Gamma	R	A.S.	-
J	Counter used to indicate point J	I	M.	-
JAND1	Counter used to indicate point J+1	I	M.	-
MACH1	Mach number at Point 1 upstream	R	M.	-
MACH2	Mach number at Point 2 downstream	R	M.	-
MACHCS	Mach number on Cone Surface	R	F.C.	-
MAXVEL	Maximum Air Velocity	R	S.A.	m/s
MFACT	Kilobar Multiplication Factor Input value for a Water run	R	I.	-
NC	Test value to determine if a complete print and/or another run is desired	C	I.	-
NUM	Value of the numerator in the calculation of Busemann Radius	DP	B.R.	m/s
ODEG	Used to output value of angle Omega in degrees	R	M.	Degrees
OMEGA	Array holding values of angle Omega	DP	M.	Radians
PKBAR	Pressure at any point in kilobars	R	M.	Kilobars
PRESCS	Pressure at Cone Surface	R	F.C.	Pascals
PRESP	User response to printout option - an Input value	C	I.	-
PRESS	Array holding values of the Pressure at each point J	DP	M.	Pascals
PRESS1	Pressure at Point 1 upstream - an Input value	DP	M.	Pascals
PRESS2	Pressure at Point 2 downstream	DP	M.	Pascals
RADIUS	Array holding values of the Busemann radius at each point J	DP	M.	m/s
RDEFL	Used to output the Deflection Angle	R	A.S.	Degrees
REFVOL	Reference Specific Volume upstream	DP	W.S. W.V.	m3/kg

TABLE VIII

Main Program Variables (cont'd.)

Variable	Variable's Meaning or Use	Type	Module	Units
SHKANG	Value of the Shock Angle - an Input value	R	M.	Degrees
SHRAD	Value of Shock Angle in Radians	R	A.S.	Radians
SPVOL2	Specific Volume at point 2	DP	W.S.	m3/kg
STEP	Step size by which Omega is decreased	R	M.	Radians
TDEG	Used for output of streamline angle	R	M.	Degrees
TEMP1	Temperature at point 1 upstream - an Input value	DP	M.	Kelvin
TEMP2	Temperature at point 2 downstream	DP	M.	Kelvin
TEMPCS	Temperature at Cone Surface	R	F.C.	Kelvin
THETA	Array holding values of the Streamline angles	DP	M.	Radians
TOTENT	Total Enthalpy	DP	S.E. W.V.	J/kg
TVALUE	Test value used to determine if cone surface has been reached	R	M.	-
U	Array holding values of Busemann velocity components in x-direction at point J	DP	B.C.	m/s
V	Array holding values of Busemann velocity components in y-direction at point J	DP	B.C.	m/s
VEL	Array holding values of the Velocity at each point J	DP	M.	m/s
VELFS	Freestream Velocity	DP	M.	m/s
VNORM1	Component of velocity Normal to the shock front upstream	DP	M.	m/s
VNORM2	Component of velocity Normal to the shock front downstream	DP	M.	m/s
VTANG	Tangential component of velocity at the shock front	DP	M.	m/s
WRESP	Test value for determining if the run is for air or water	C	I.	-
X	Array holding values of the x-coord. of the Busemann circle center	DP	B.C.	m/s

TABLE IX

Main Program Variables (cont'd.)

Variable	Variable's Meaning or Use	Type	Module	Units
Y	Array holding values of the y-coord. of the Busemann circle center	DP	B.C.	m/s
YES	Test value used to determine if complete print and/or another run is desired	C	I.	-

LEGEND

(1) TYPE is the type of data represented with designations as follows:

R = Real Single Precision
DP = Real Double Precision
I = Integer (used for counters)
L = Logical (i.e. True or False)
C = Character (used to store character information - is an integer type in FORTRAN)

(2) MODULE is the primary, but not exclusive, location of use of the variable. The abbreviations used mean:

A.S. = Airshock Calculation Section of Main Program
B.C. = Busemann Circle Center Calculation Section of Main Program
B.R. = Busemann Radius Calculation Section of Main Program
I. = Input Section of Main Program
F.C. = Final Output Section of Main Program
M. = Variable Used Throughout Main Program
S.A. = Parameter to SOSAIR Module
S.E. = Stagnation Enthalpy Calculation of Main Program
W.S. = Parameter to WSHOCK Module
W.V. = Parameter to WATVEL Module

Note: this legend applies to all Tables listing program variables.

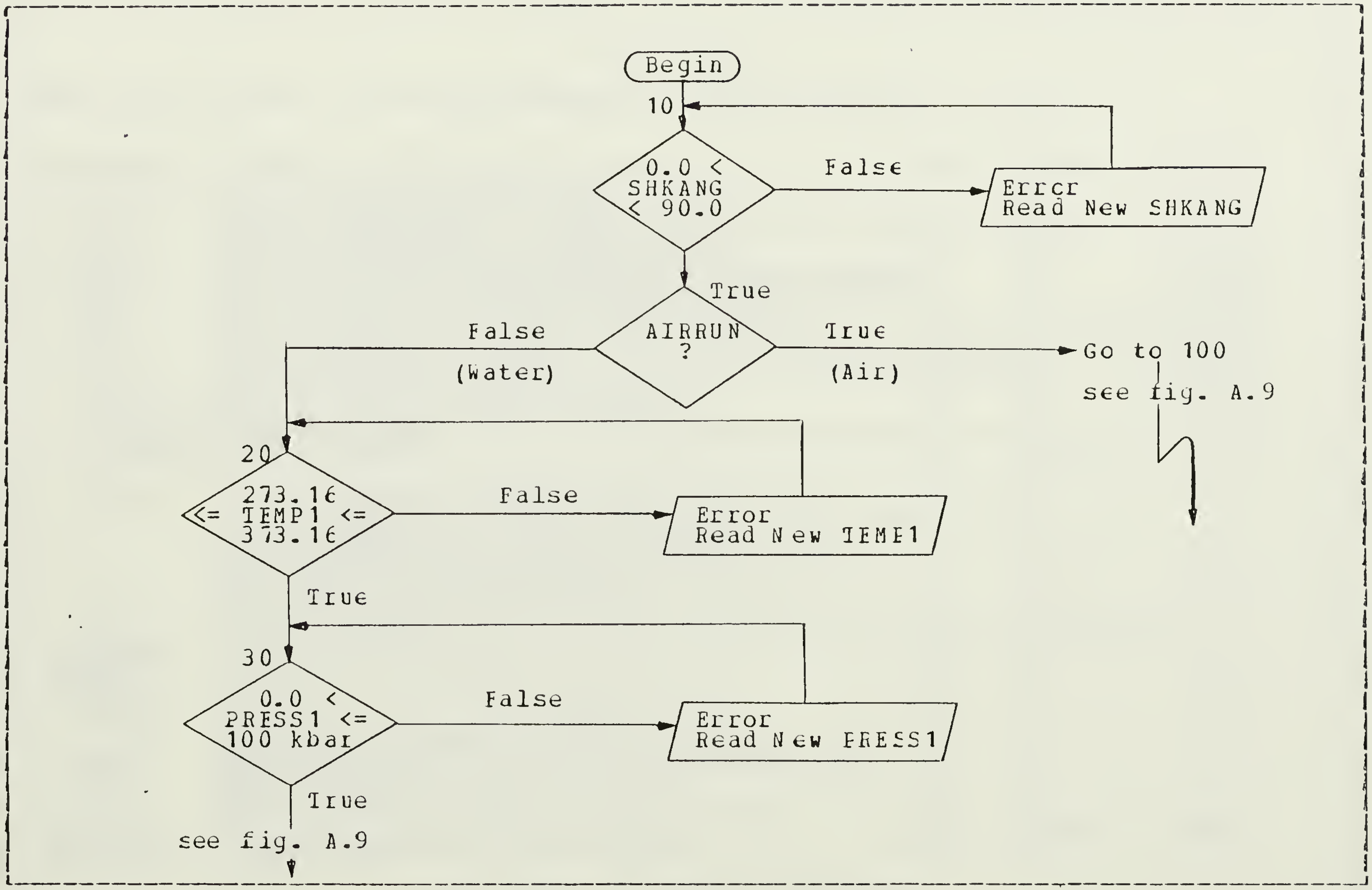

Figure A.8 Subroutine CHKINP Flowchart.

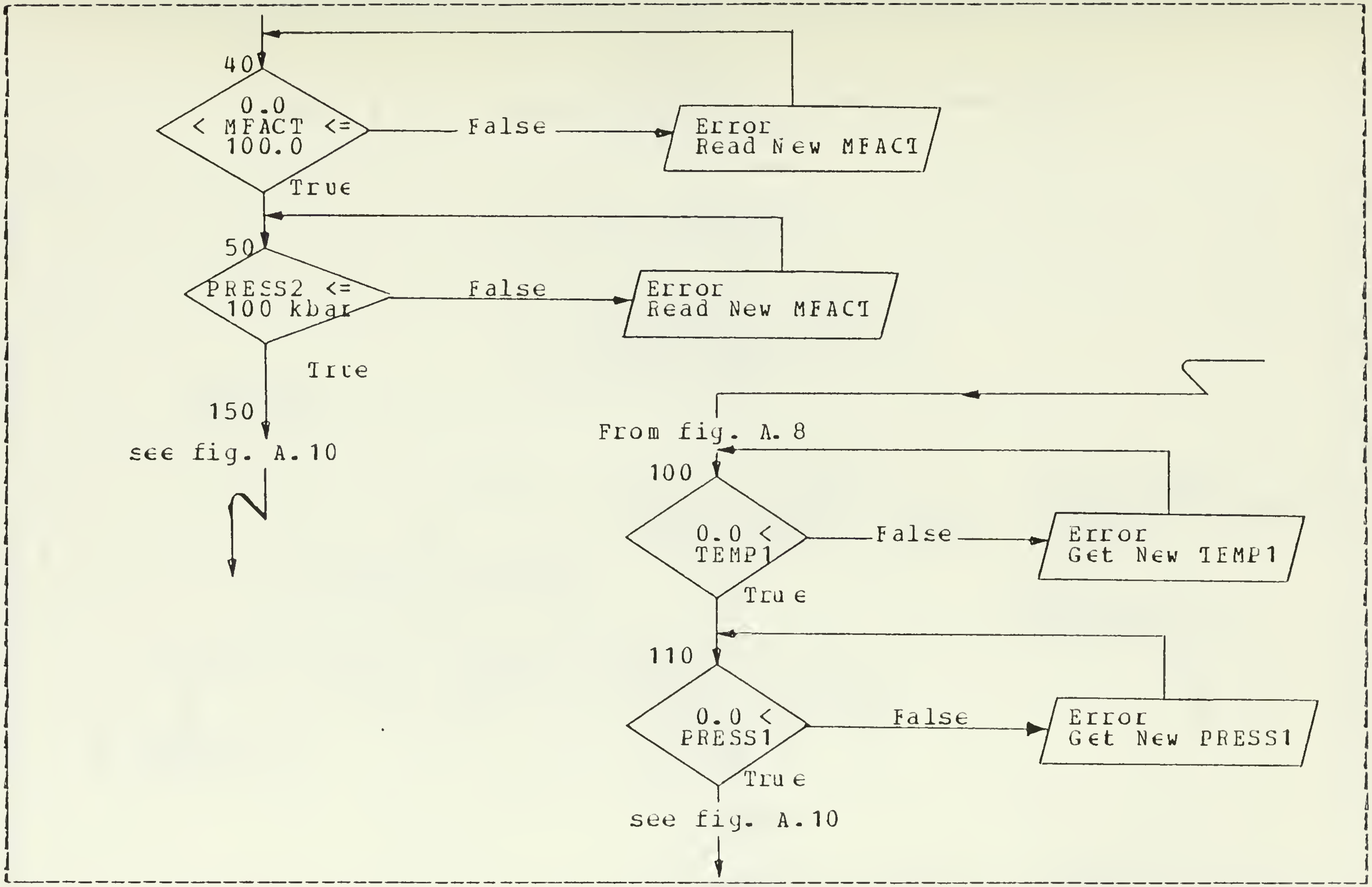

Figure A.9 Subroutine CHKINP Flowchart (cont'd.).

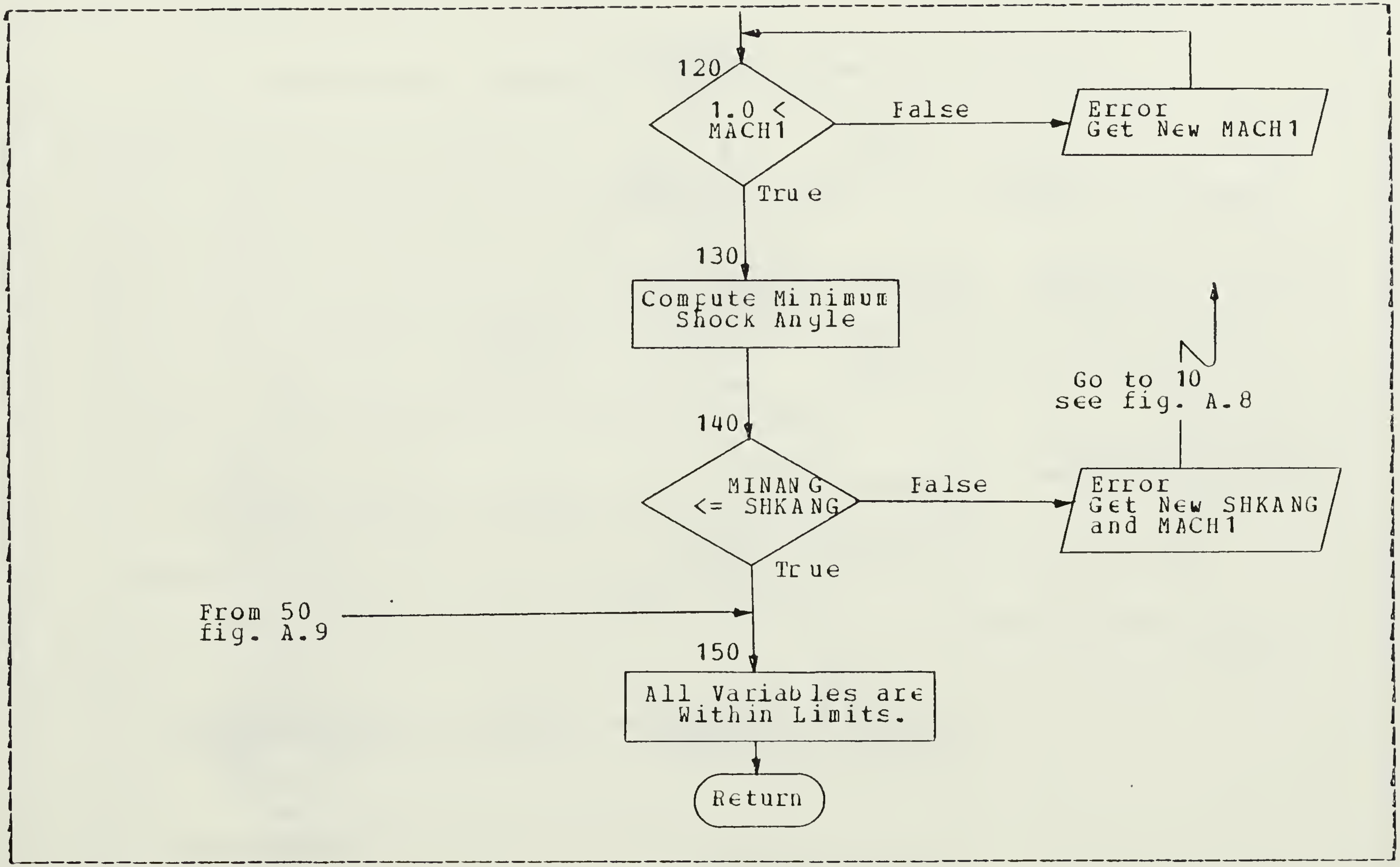

Figure A.10 Subroutine CHKINP Flowchart (cont'd.).

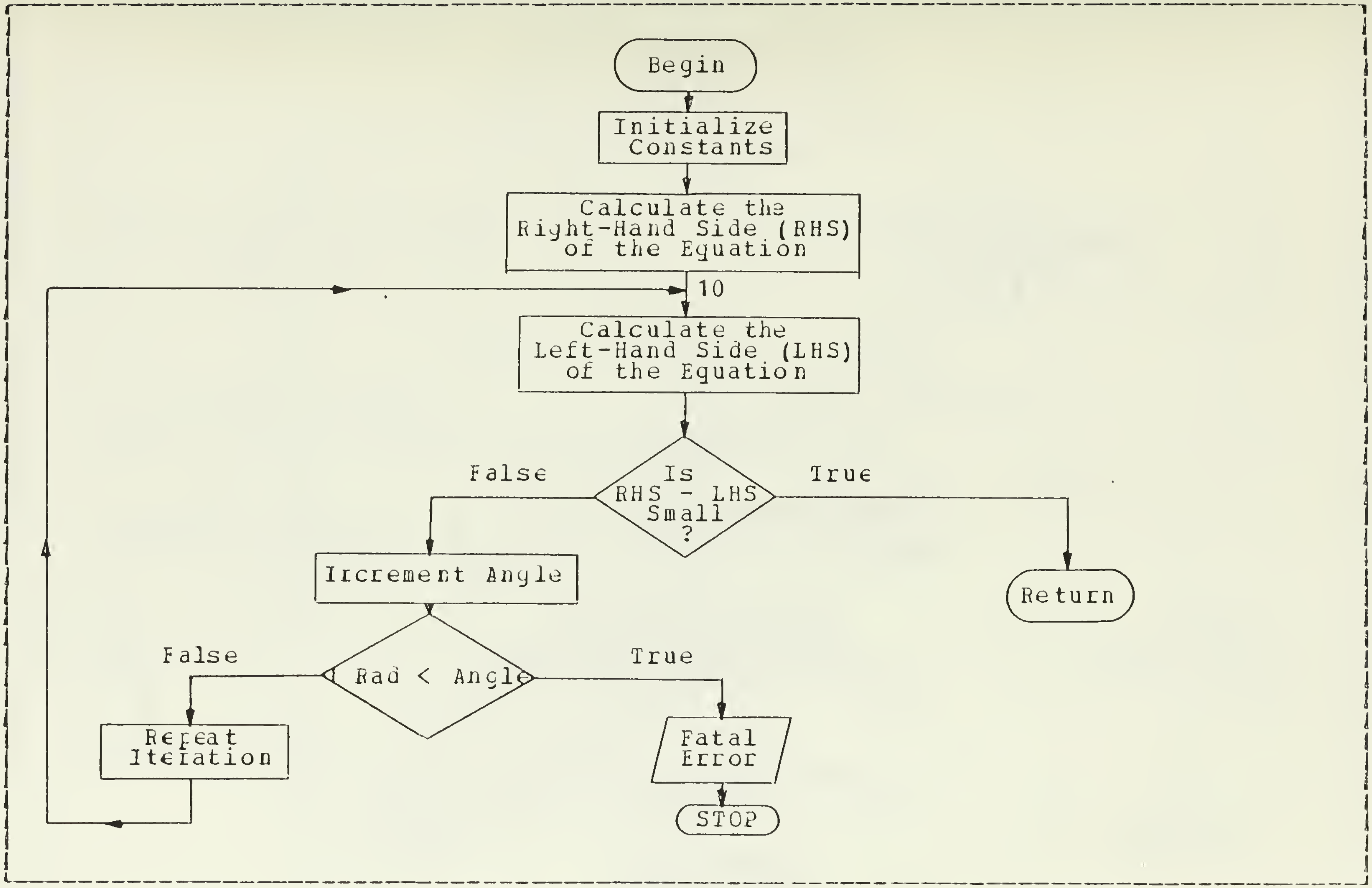

Figure A.11 Subroutine DEFANG Flowchart.

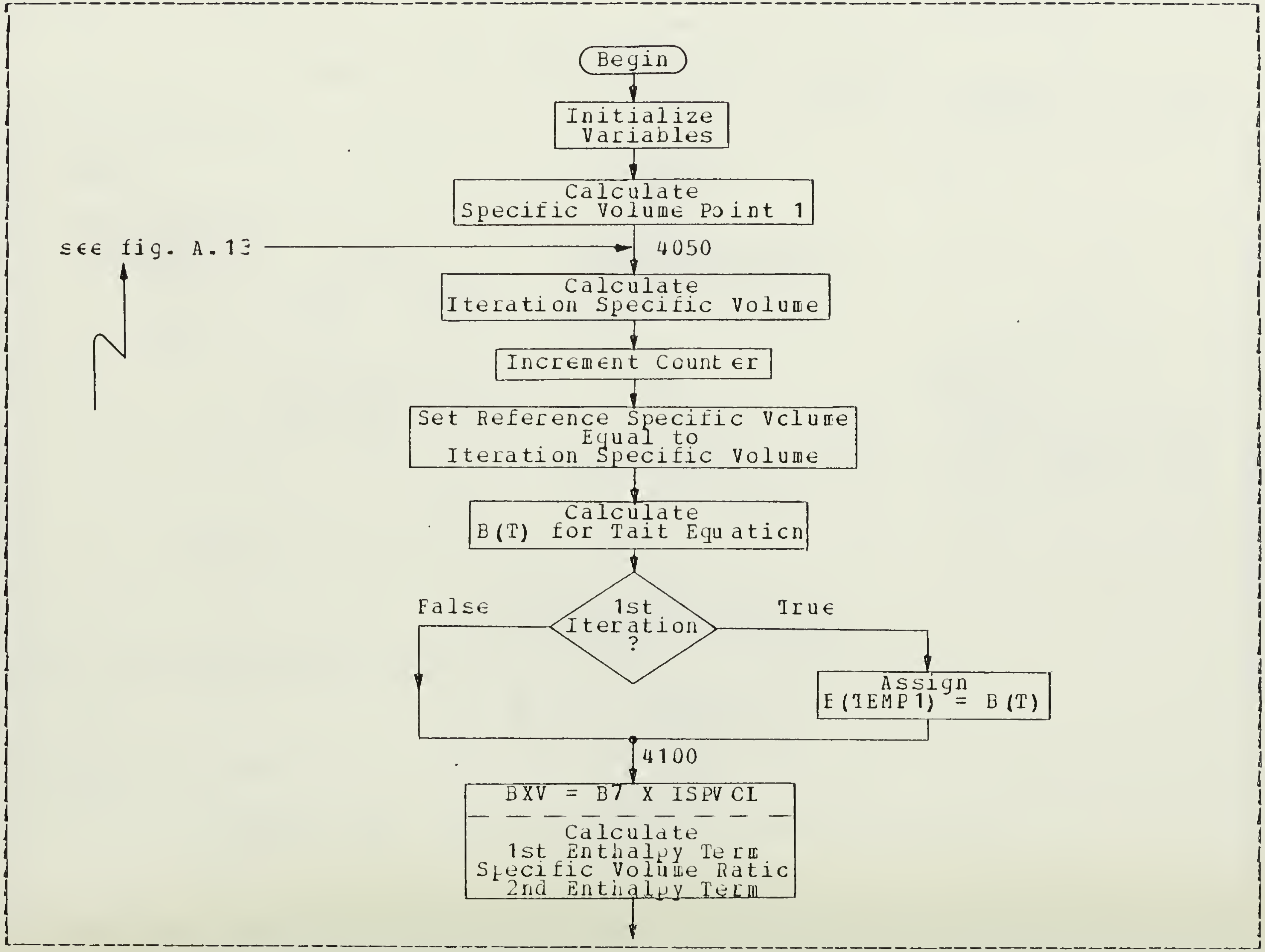
Begin
Initialize Variables
Calculate Specific Volume Point 1
see fig. A.13
4050
Calculate Iteration Specific Volume
Increment Counter
Set Reference Specific Volume Equal to Iteration Specific Volume
Calculate B(T) for Tait Equation
1st Iteration ?
False
True
Assign E(TEMP1) = B(T)
4100
BXV = B7 X ISPVCL
Calculate 1st Enthalpy Term Specific Volume Ratio 2nd Enthalpy Term

Figure A.12 Subroutine WSHOCK Flowchart.

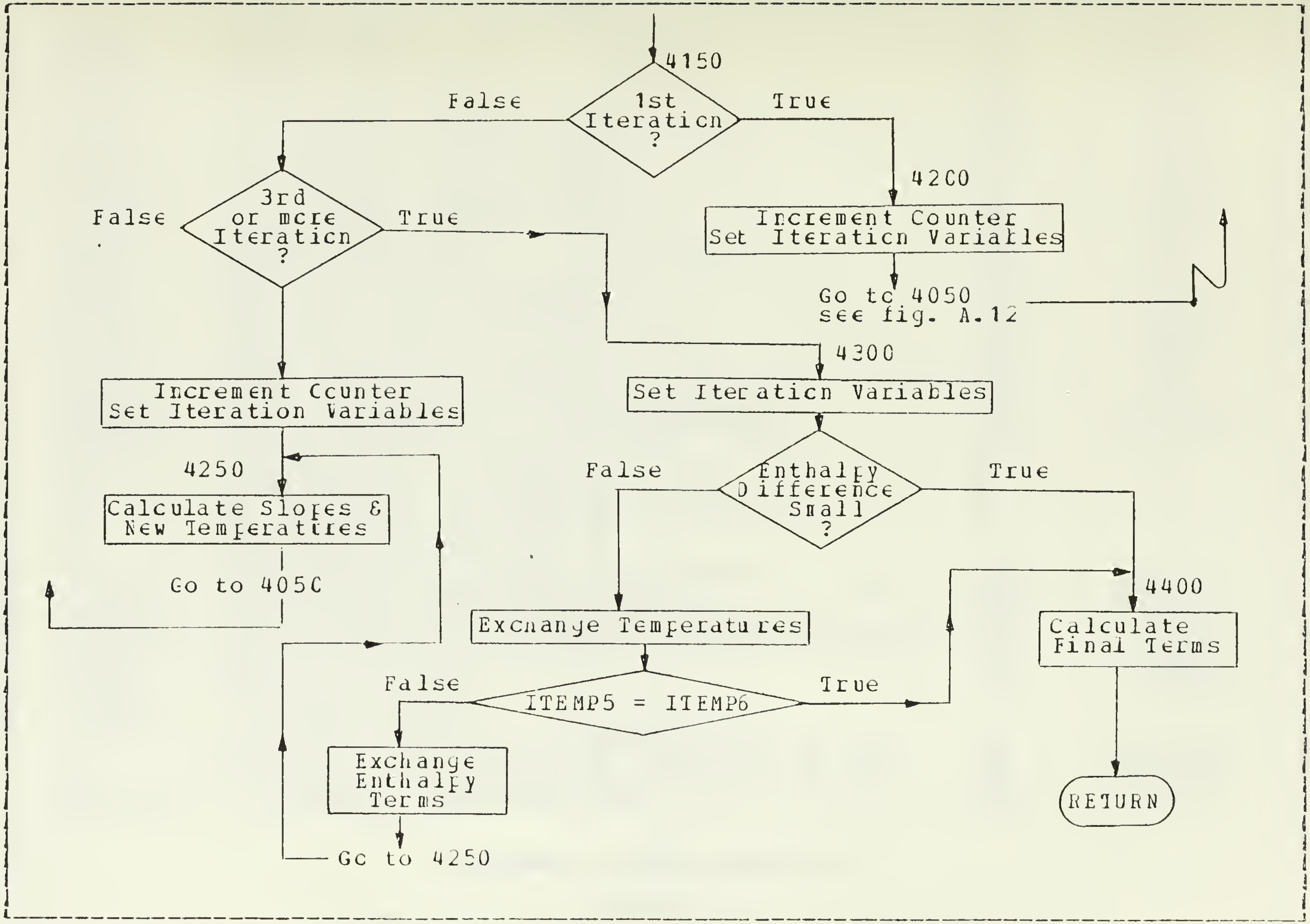

Figure A.13 Subroutine WSHOCK Flowchart (cont'd.).

TABLE X

Subroutine WSHOCK Variables

Variable	Variable's Meaning or Use	Type	Units
E6	A constant used in the Tait equation of EPV	DP	Pascals
E7	A constant used in the Tait equation of EPV	DP	Pascals
EXV	Product of B(T) times SPVOL(T)	DP	m2/s2
COUNT1	A program counter	I	-
COUNT2	A program counter	I	-
DENS1	Density at Point 1	DP	kg/m3
DENS2	Density at Point 2	DP	kg/m3
ISPVOL	Iteration Specific Volume	DP	m3/kg
ITEMP1	An iteration temperature	DP	Kelvin
ITEMP2	An iteration temperature	DP	Kelvin
ITEMP3	An iteration temperature	DP	Kelvin
N	A constant used in the Tait equation	R	-
N1	An expression involving N	R	-
N2	An expression involving N	R	-
N2	An expression involving N	R	-
N3	An expression involving N	R	-
N4	An expression involving N	R	-
REFSCS	Reference Speed of Sound	DP	m/s
SLOPE1	Slope of WSENT9 vs. temperature curve	DP	J/Kelvin
SLOPE2	Slope of WSENT8 vs. temperature curve	DP	J/Kelvin
SCS1	Speed of Sound at Point 1	DP	m/s
SCS2	Speed of Sound at Point 2	DP	m/s
SPVOL0	Reference Specific Volume	DP	m3/kg
SPVOL1	Specific Volume at Point 1	DP	m3/kg
SVOL2	Specific Volume at Point 2	DP	m3/kg
SVRAT	Ratio of specific volumes	DP	-
VEL1	Velocity at Point 1	DP	m/s
VEL2	Velocity at Point 2	DP	m/s
WSENT2	An iteration enthalpy term	DP	J/kg
WSENT3	An iteration enthalpy term	DP	J/kg
WSENT4	An iteration enthalpy term	DP	J/kg
WSENT5	An iteration enthalpy term	DP	J/kg
WSENT6	An iteration enthalpy term	DP	J/kg
WSENT7	An iteration enthalpy term	DP	J/kg
WSENT8	An iteration enthalpy term	DP	J/kg
WSENT9	An iteration enthalpy term	DP	J/kg
WSPRES	Pressure at Point 2	DP	Pascals
WSTEST	A test value	R	-
WTEMP1	Temperature at Point 1	DP	Kelvin
WVEL2	Velocity of the Shock Front	DP	m/s

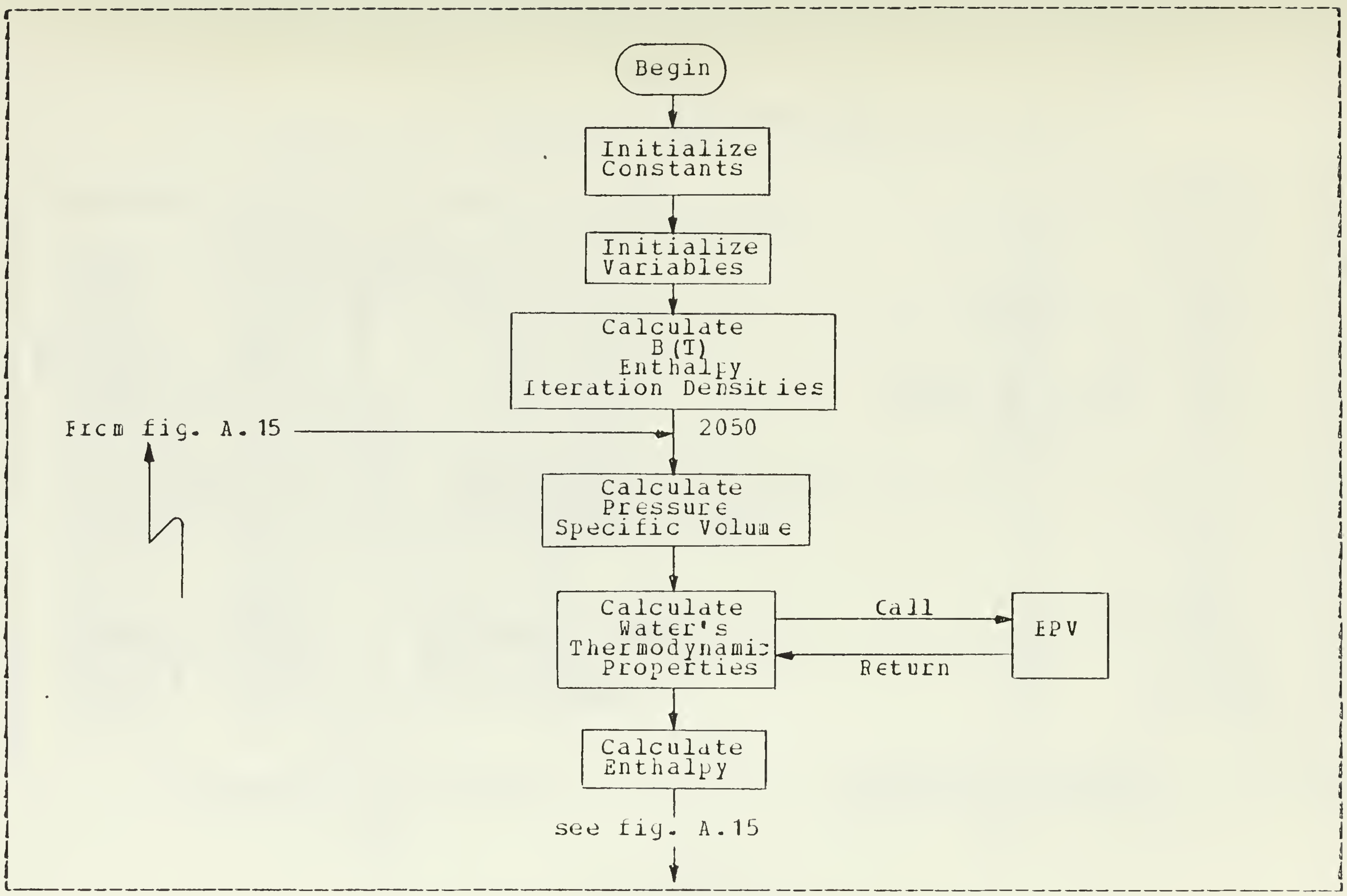
Begin
Initialize Constants
Initialize Variables
Calculate B(T) Enthalpy Iteration Densities
From fig. A.15
2050
Calculate Pressure Specific Volume
Calculate Water's Thermodynamic Properties
Call
EPV
Return
Calculate Enthalpy
see fig. A.15

Figure A.14 Subroutine WATVEL Flowchart.

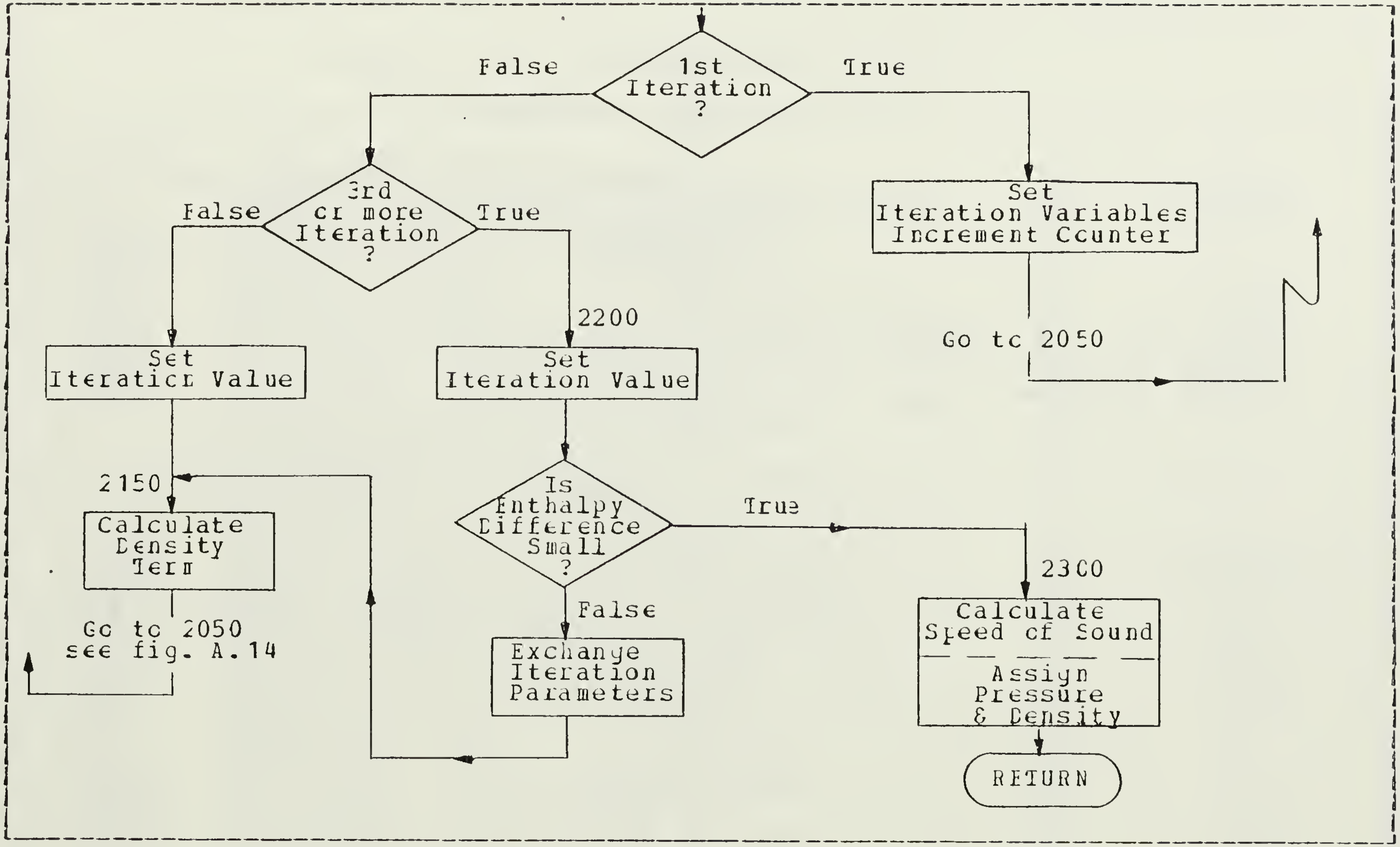

Figure A.15 Subroutine WATVEL Flowchart (cont'd.).

TABLE XI

Subroutine WATVEL Variables

Variable	Variable's Meaning or Use	Type	Units
E	Constant used in Tait equation of EPV	DP	Pascals
COUNT	Program counter	I	-
DENS3	An iteration density term	DP	kg/m3
DENS4	An iteration density term	DP	kg/m3
DENS5	An iteration density term	DP	kg/m3
DENSJ	Density at Point J	DP	kg/m3
DENSJ1	Density at Point J+1	DP	kg/m3
ENTHAL	Enthalpy	DP	J/kg
ENTHE1	An enthalpy iteration term	DP	J/kg
ENTHE2	An enthalpy iteration term	DP	J/kg
ENTHE3	An enthalpy iteration term	DP	J/kg
N	Constant used in the Tait equation	R	-
N1	An expression involving N	R	-
PRESJ1	Pressure at Point J+1	DP	Pascals
REFSPD	Speed of Sound at Point 1	DP	m/s
RVOL	Reference Specific Volume at Point 1	DP	m3/kg
SENTH	Stagnation Enthalpy	DP	J/kg
SOSJ1	Speed of Sound at Point J+1	DP	m/s
VELJ	Velocity at Point J	DP	m/s
WDENS	Density at Point J+1	DP	kg/m3
WENERG	Energy term returned by EPV	DP	J/kg
WENTH	Enthalpy at Point J+1	DP	J/kg
WPRESS	Pressure at Point J+1	DP	Pascals
WSPVOL	Specific Volume at Point J+1	DP	m3/kg
WTEMP	Temperature at Point J+1	DP	Kelvin
WTEST	Test value	R	-

APPENDIX B

SAMPLE PRINTOUTS

This appendix presents copies of the output from various runs of the computer program listed in Appendix C. The various samples illustrate the output from the following options:

(1) A Complete Print of a Water Run

(2) A Summary Print of a Water Run

(3) A Complete Print of an Air Run, and

(4) A Summary Print of an Air Run

(Note: in order to reduce the volume of the thesis, only a portion of the output from the complete print options is included.)

SAMPLE 1 - COMPLETE PRINTOUT (WATER CASE)

```
*** THIS RUN IS FOR  - WATER - ***

   INPUT VALUES WERE AS FOLLOWS:

UPSTREAM TEMPERATURE =  323.16  KELVIN
UPSTREAM PRESSURE =          100000.0  PASCALS
SHOCK ANGLE =  20.0  DEGREES
THE PRESSURE MULTIPLICATION FACTOR (MFACT) =  5.0

   CONDITIONS CALCULATED FOR POINT 1 UPSTREAM:

REFERENCE SPECIFIC VOLUME =         0.00100411  M3/KG
DENSITY AT POINT 1 =         996.5139  KG/M3

NORMAL COMPONENT OF VELOCITY =      2017.5467  M/S
TANGENTIAL COMPONENT OF VELOCITY =      5543.1641  M/S
FREESTREAM VELOCITY =       5898.9120  M/S

REFERENCE SPEED OF SOUND =      1510.7939  M/S
SPEED OF SOUND AT POINT 1 =      1510.5239  M/S

FREESTREAM MACH NUMBER =         3.9052


   CONDITIONS CALCULATED FOR POINT 2 DOWNSTREAM

PRESSURE AT POINT 2 =        500000000.0  PASCALS
   THIS PRESSURE IN KILOBARS =    5.0 KILOBARS
SPECIFIC VOLUME AT POINT 2 =   0.00087980  M3/KG
DENSITY AT POINT 2 =        1136.6191 KG/M3

EEAR (FROM EPV) AT POINT 2 =      30923.2949  J/KG
ENTHALPY AT POINT 2 =    470824.3162  J/KG
STAGNATION ENTHALPY =  17398580.6611  J/KG

NORMAL COMPONENT OF VELOCITY =      1768.8540  M/S
VELOCITY AT POINT 2 =        5818.5490  M/S
WATER VELOCITY AT POINT 2 =   248.6927 M/S

SPEED OF SOUND AT POINT 2 =       2268.3064  M/S
MACH NUMBER AT POINT 2 =          2.5652

X-COMPONENT OF BUSEMANN VELOCITY =       5813.8541  M/S
Y-COMPONENT OF BUSEMANN VELOCITY =        233.6947  M/S
```

```
OMEGA AT POINT 2 =   20.0  DEGREES
STREAMLINE ANGLE AT POINT 2 =    2.3018  DEGREES

BUSEMANN RADIUS AT POINT  2 =       1743.5342   M/S
X-COORDINATE OF BUSEMANN CENTER =         7452.2403   M/S
Y-COORDINATE OF BUSEMANN CENTER =          830.0185  M/S


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

   POINT =   3

X-COMPONENT OF VELOCITY AT POINT  3 =       5803.6963  M/S
Y-COMPONENT OF VELOCITY AT POINT  3 =        262.3793  M/S
VELOCITY AT POINT  3 =       5809.6243  M/S

OMEGA AT POINT  3 =   19.0  DEGREES
STREAMLINE ANGLE AT POINT  3 =    2.5885  DEGREES

ENTHALPY AT POINT  3 =     522713.5817       J/KG
PRESSURE AT POINT  3 =       684748052.6934    PASCAL
   THIS PRESSURE IN KILOBARS =    6.8475 KILOBARS
DENSITY AT POINT  3 =       1183.3863   KG/M3
SPEED OF SOUND AT POINT  3 =       2567.7326   M/S

BUSEMANN RADIUS AT POINT  3 =       1362.8023   M/S
X-COORDINATE OF BUSEMANN CENTER =         7092.2512   M/S
Y-COORDINATE OF BUSEMANN CENTER =          706.0643  M/S


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

CONE HAS BEEN REACHED - FOLLOWING IS A SUMMARY OF RESULTS:

CONE SEMI-VERTEX ANGLE =      7.2299  DEGREES

SHOCK ANGLE =    20.0  DEGREES
FREESTREAM MACH NUMBER =      3.9052
FREESTREAM VELOCITY =      5898.9120  M/S

VELOCITY AT CONE SURFACE =        5753.2645   M/S
SPEED OF SOUND AT CONE SURFACE =        2987.4511  M/S
MACH NUMBER AT CONE SURFACE =           1.9258

PRESSURE AT CONE SURFACE =      1092461570.0          PASCALS
   PRESSURE AT CONE SURFACE =   10.9246  KILOBARS
ENTHALPY AT CONE SURFACE =    848554.2500         J/KG
```

```
DENSITY AT CONE SURFACE =        1243.1082            KG/M3
DRAG COEFFICIENT (CD) =           0.0630

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

SAMPLE 2 - SUMMARY PRINTOUT (WATER CASE)

```
*** THIS RUN IS FOR  - WATER - ***

   INPUT VALUES WERE AS FOLLOWS:

UPSTREAM TEMPERATURE =  298.16  KELVIN
UPSTREAM PRESSURE =         101300.0  PASCALS
SHOCK ANGLE =  25.0  DEGREES
THE PRESSURE MULTIPLICATION FACTOR (MFACT) = 10.0

   CONDITIONS CALCULATED FOR POINT 1 UPSTREAM:

REFERENCE SPECIFIC VOLUME =           0.00099599  M3/KG
DENSITY AT POINT 1 =       1005.8843  KG/M3

NORMAL COMPONENT OF VELOCITY =        2358.1095  M/S
TANGENTIAL COMPONENT OF VELOCITY =       5056.9817  M/S
FREESTREAM VELOCITY =       5579.7621  M/S

REFERENCE SPEED OF SOUND =       1494.8674  M/S
SPEED OF SOUND AT POINT 1 =       1485.0679  M/S

FREESTREAM MACH NUMBER =         3.7572


   CONDITIONS CALCULATED FOR POINT 2 DOWNSTREAM

PRESSURE AT POINT 2 =      1013000000.0   PASCALS
   THIS PRESSURE IN KILOBARS =   10.13  KILOBARS
SPECIFIC VOLUME AT POINT 2 =    0.00081410  M3/KG
DENSITY AT POINT 2 =       1228.3457 KG/M3

HEAT (FROM EPV) AT POINT 2 =      91194.8388  J/KG
ENTHALPY AT POINT 2 =     915881.2343  J/KG
STAGNATION ENTHALPY =  15566873.3346  J/KG

NORMAL COMPONENT OF VELOCITY =        1931.0411  M/S
VELOCITY AT POINT 2 =       5413.1307  M/S
WATER VELOCITY AT POINT 2 =   427.0684 M/S

SPEED OF SOUND AT POINT 2 =       2779.0381  M/S
MACH NUMBER AT POINT 2 =          1.9478

X-COMPONENT OF BUSEMANN VELOCITY =       5399.2752   M/S
Y-COMPONENT OF BUSEMANN VELOCITY =        387.0554  M/S
```

```
OMEGA AT POINT 2 =   25.0000  DEGREES
STREAMLINE ANGLE AT POINT 2 =     4.1003  DEGREES

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ** *

CONE HAS BEEN REACHED - FOLLOWING IS A SUMMARY OF RESULTS:

CONE SEMI-VERTEX ANGLE =    10.5470  DEGREES

SHOCK ANGLE =    25.0  DEGREES
FREESTREAM MACH NUMBER =     3.7572
FREESTREAM VELOCITY =      5579.7621  M/S

VELOCITY AT CONE SURFACE =        5316.8138   M/S
SPEED OF SOUND AT CONE SURFACE =        3534.3125   M/S
MACH NUMBER AT CONE SURFACE =           1.5041

PRESSURE AT CONE SURFACE =      1800928000.0           PASCALS
   PRESSURE AT CONE SURFACE =   18.0093 KILOBARS
ENTHALPY AT CONE SURFACE =   1432619.0  J/KG
DENSITY AT CONE SURFACE =       1328.2969  KG/M3
DRAG COEFFICIENT (CD) =          0.1150

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  ** * *
```

SAMPLE 3 - COMPLETE PRINTOUT (AIR CASE)

```
*** THIS RUN IS FOR  - AIR - ***

   INPUT VALUES WERE AS FOLLOWS:

UPSTREAM TEMPERATURE =   300.16  KELVIN
UPSTREAM PRESSURE =          100000.0  PASCALS
SHOCK ANGLE =  30.0  DEGREES
FREESTREAM MACH NUMBER =          3.0

   CALCULATED INITIAL CONDITIONS WERE:

DEFLECTION ANGLE =   12.7735  DEGREES
DENSITY AT POINT 1 =          1.1610  KG/M3

FREESTREAM VELOCITY =       1041.7729  M/S
MAXIMUM AIR VELOCITY = 1299.3167  M/S

MACH NUMBER AT POINT 2 =          2.3673
VELOCITY AT POINT 2 =        944.5709  M/S
TANGENTIAL COMPONENT OF VELOCITY =        902.1986  M/S

X-COMPONENT OF BUSEMANN VELOCITY =        921.1945  M/S
Y-COMPONENT OF BUSEMANN VELOCITY =        208.8416  M/S

OMEGA AT POINT 2 =   30.0  DEGREES
STREAMLINE ANGLE AT POINT 2 =   12.7735  DEGREES

BUSEMANN RADIUS AT POINT  2 =       821.4434  M/S
X-COORDINATE OF BUSEMANN CENTER =       1632.5854  M/S
Y-COORDINATE OF BUSEMANN CENTER =        619.5633  M/S


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

   POINT =  3

X-COMPONENT OF VELOCITY AT POINT  3 =       914.1348  M/S
Y-COMPONENT OF VELOCITY AT POINT  3 =       221.3196  M/S
VELOCITY AT POINT  3 =        940.5449  M/S

OMEGA AT POINT  3 =   29.0  DEGREES
STREAMLINE ANGLE AT POINT  3 =   13.6099  DEGREES

SPEED OF SOUND AT POINT  3 =        400.8987  M/S
```

```
BUSEMANN RADIUS AT POINT  3 =       745.5216  M/S
X-COORDINATE OF BUSEMANN CENTER =      1566.1827  M/S
Y-COORDINATE OF BUSEMANN CENTER =       582.7557  M/S

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

CONE HAS BEEN REACHED - FOLLOWING IS A SUMMARY OF RESULTS:

CONE SEMI-VERTEX ANGLE =    20.3996  DEGREES

SHOCK ANGLE =    30.0  DEGREES
FREESTREAM MACH NUMBER =     3.0
FREESTREAM VELOCITY =       1041.7729  M/S

VELOCITY AT CONE SURFACE =        925.8313  M/S
SPEED OF SOUND AT CONE SURFACE =        407.6909  M/S
MACH NUMBER AT CONE SURFACE =          2.2709

PRESSURE AT CONE SURFACE =          285849.8120  PASCALS
TEMPERATURE AT CONE SURFACE =        413.7244  KELVIN
DENSITY AT CONE SURFACE =          2.4077   KG/M3
DRAG COEFFICIENT (CD) =          0.2950

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

SAMPLE 4 - SUMMARY PRINTOUT (AIR CASE)

```
*** THIS RUN IS FOR  - AIR - ***

   INPUT VALUES WERE AS FOLLOWS:

UPSTREAM TEMPERATURE =  373.16  KELVIN
UPSTREAM PRESSURE =          202600.0  PASCALS
SHOCK ANGLE =  45.0  DEGREES
FREESTREAM MACH NUMBER =          3.25

   CALCULATED INITIAL CONDITIONS WERE:

DEFLECTION ANGLE =   27.0239  DEGREES
DENSITY AT POINT 1 =        1.8920  KG/M3

FREESTREAM VELOCITY =      1258.3641  M/S
MAXIMUM AIR VELOCITY = 1527.4334  M/S

MACH NUMBER AT POINT 2 =         1.7324
VELOCITY AT POINT 2 =        935.4593  M/S
TANGENTIAL COMPONENT OF VELOCITY =       889.7951  M/S

X-COMPONENT OF BUSEMANN VELOCITY =       833.3229  M/S
Y-COMPONENT OF BUSEMANN VELOCITY =       425.0375  M/S

OMEGA AT POINT 2 =   45.0  DEGREES
STREAMLINE ANGLE AT POINT 2 =   27.0239  DEGREES


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

CONE HAS BEEN REACHED - FOLLOWING IS A SUMMARY OF RESULTS:

CONE SEMI-VERTEX ANGLE =    35.3971  DEGREES

SHOCK ANGLE =    45.0  DEGREES
FREESTREAM MACH NUMBER =     3.25
FREESTREAM VELOCITY =       1258.3641  M/S

VELOCITY AT CONE SURFACE =        914.1494  M/S
SPEED OF SOUND AT CONE SURFACE =         547.2446  M/S
MACH NUMBER AT CONE SURFACE =          1.6705

PRESSURE AT CONE SURFACE =        1333402.0  PASCALS
TEMPERATURE AT CONE SURFACE =        745.4390  KELVIN
```

```
DENSITY AT CONE SURFACE =          6.2334  KG/M3
DRAG COEFFICIENT (CD) =          0.7549

. * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

APPENDIX C

PROGRAM LISTING

This appendix contains a complete listing of the fully documented main program named CONEFLOW and all of its functions and subroutines. The functions and subroutines included in this appendix are as follows:

(1) Function DTOR

(2) Function RTOD

(3) Function DEFANG

(4) Function SOSAIR

(5) Subroutine CERINP

(6) Subroutine WSHOCK

(7) Subroutine WATVEL

(8) Subroutine EPV

```
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                         PROGRAM CONEFLOW
C
C
C  CREATED BY:   LT PATRICK W. HUGHES, U.S. NAVY
C                NAVAL POSTGRADUATE SCHOOL
C                MONTEREY, CA.  93940
C
C
C  ORIGINAL PROGRAM CREATED:  1 JUNE 1984
C  MODIFICATIONS/DATES:  NO MODIFICATIONS TO DATE
C
C
C  PROGRAM PURPOSE:  THIS PROGRAM WAS CREATED TO CALCULATE THE THERMO-
C                    DYNAMIC QUANTITIES NEEDED TO PREDICT THE HYDRODY-
C                    NAMIC FLOW OVER A CONE. THE PROGRAM WAS DESIGNED TO
C                    GIVE RESULTS FOR FLOW IN EITHER AN AIR OR A WATER
C                    FLUID. THE VARIABLE ULTIMATELY DETERMINED BY THIS
C                    PROGRAM IS THE CONE SEMI-VERTEX ANGLE.
C
C
C  DESCRIPTION OF ALGORITHM:  FOR A COMPLETE LOGIC DESCRIPTION OF THE
C                    PROGRAM, THE USER SHOULD REFER TO THE FLOWCHARTS.
C                    A GENERAL OUTLINE OF THE ALGORITHM FOLLOWED IN THE
C                    COMPUTATIONS IS PRESENTED BELOW.
C
C                      READ USER INPUT
C                      VERIFY USER INPUT IS VALID
C                      IF INPUT VALID THEN
C                        CONTINUE CALCULATIONS
C                      ELSE
C                        GET NEW INPUT FROM USER
C                        VERIFY NEW INPUT
C
C                      CALCULATE INITIAL THERMODYNAMIC CONDITIONS ON
C                           BOTH SIDES OF THE SHOCK FRONT
C                      ITERATE ALONG STREAMLINES USING BUSEMANN GRAPH-
C                           ICAL PROCEDURE UNTIL CONE SURFACE IS REACHED
C                      CALCULATE CONDITIONS ON CONE SURFACE
C                      PRINT FINAL RESULTS
C                      REPEAT
C                        IF USER DESIRES
C                      ELSE
C                        STOP
C
C
C
```

```
C                    THE PROGRAM REQUIRES THAT THE USER SUPPLY THE FOL-
C                    LOWING INPUT:
C
C                      (1) AN INDICATION OF WHETHER A COMPLETE OR A
C                          SUMMARY PRINTOUT IS DESIRED. THIS IS RECEIVED
C                          BY THE PROGRAM THROUGH VARIABLE -PRESP-.
C
C                      (2) AN INDICATION OF WHETHER THIS RUN IS FOR AIR
C                          OR FOR WATER. THIS IS RECEIVED BY THE PROGRAM
C                          THROUGH VARIABLE -AWRESP-.
C
C                      (3) A VALUE FOR THE TEMPERATURE UPSTREAM OF THE
C                          SHOCK FRONT. THIS IS RECEIVED BY THE PROGRAM
C                          THROUGH VARIABLE -TEMP1-.
C
C                      (4) A VALUE FOR THE PRESSURE UPSTREAM OF THE
C                          SHOCK FRONT. THIS IS RECEIVED BY THE PROGRAM
C                          THROUGH VARIABLE -PRESS1-.
C
C                      (5) A VALUE FOR THE ANGLE THE SHOCK FRONT MAKES
C                          WITH THE HORIZONTAL. THIS IS RECEIVED BY THE
C                          PROGRAM THROUGH VARIABLE -SHKANG-.
C
C                      (6) IF THE RUN IS FOR AIR, A VALUE FOR THE MACH
C                          NUMBER UPSTREAM OF THE SHOCK FRONT. THIS IS
C                          RECEIVED BY THE PROGRAM THROUGH VARIABLE
C                          -MACH1-.
C
C                      (7) IF THE RUN IS FOR WATER, A VALUE FOR THE
C                          PRESSURE MULTIPLICATION FACTOR USED TO CON-
C                          VERT TO A PRESSURE, AT POINT 2 DOWNSTREAM OF
C                          THE SHOCK FRONT, WHICH IS PRESS1 TIMES MFACT
C                          KILOBARS. THIS IS RECEIVED BY THE PROGRAM
C                          THROUGH VARIABLE -MFACT-.
C
C
C  LIST OF PROCEDURES:  IN ADDITION TO STANDARD FUNCTIONS AND/OR PROCE-
C                    DURES (SUCH AS SIN AND COS) FOUND IN LIBRARIES,
C                    THIS PROGRAM UTILIZES THE FOLLOWING FUNCTIONS AND
C                    SUBROUTINES:
C
C                     DTOR   - A FUNCTION
C                     RTOD   - A FUNCTION
C                     DEFANG - A FUNCTION
C                     SOSAIR - A FUNCTION
C                     CHKINP - A SUBROUTINE
C                     WSHOCK - A SUBROUTINE
C                     WATVEL - A SUBROUTINE
```

```
C                      EPV    - A SUBROUTINE
C
C                    EACH OF THESE FUNCTIONS OR SUBROUTINES, EXCEPT EPV,
C                    IS EXPLAINED IN DETAIL BELOW. SUBROUTINE EPV IS
C                    INCLUDED TO SHOW HOW IT WAS TRANSLATED TO THE
C                    FORTRAN LANGUAGE AND FOR USE BY THIS PROGRAM. IT IS
C                    COPIED, BY PERMISSION OF THE AUTHOR, FROM NAVAL
C                    POSTGRADUATE SCHOOL REPORT NPS67-82-001 BY
C                    A. E. FUHS. THE SUBROUTINE EPV IS ADEQUATELY DES-
C                    CRIBED AND FLOWCHARTED IN THAT WORK AND NEED NOT BE
C                    REPEATED HERE.
C
C
C  IMPLEMENTATION NOTES:  THERE ARE NO IMPLEMENTATION SPECIFIC FEATURES
C                    OF THE NAVAL POSTGRADUATE SCHOOL COMPUTER SYSTEM
C                    INCLUDED IN THIS PROGRAM.
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C                        MAIN PROGRAM CONEFLOW
C
C  DECLARE MAIN PROGRAM VARIABLES
C
      REAL AIRM, AIRR, ANGDEF, ARGUM1, ARGUM2, CONANG, DENSCS, DRAGCO,
     *     ENTHCS, GAMMA, GAMMA1, GAMMA2, GAMMA3, MACH1, MACH2, MACHCS,
     *     MAXVEL, MFACT, PDEG, PKBAR, PRESCS, RDEFL, SHKANG, SHRAD,
     *     STEP, TDEG, TEMPCS, TVALUE, WATSPD
C
      DOUBLE PRECISION C, CREF, DEN, DENSTY, DENSY1, DENSY2, ENERG,
     *                 ENTH2, ENTHLP, NUM, OMEGA, PRESS, PRESS1, PRESS2,
     *                 RADIUS, REFVOL, SPVOL2, TEMP1, TEMP2, THETA,
     *                 TOTENT, U, V, VEL, VELPS, VNORM1, VNORM2, VTANG,
     *                 X, Y
C
      LOGICAL AIRRUN, CONERT
C
C  THE FOLLOWING INTEGER VARIABLES ARE USED AS COUNTERS
C
      INTEGER J, JAND1
C
C  THE FOLLOWING INTEGER VARIABLES ARE USED TO STORE CHARACTERS
C
      INTEGER AGAIN, ARESP, AWRESP, NO, PRESP, WRESP, YES
C
C
C  ESTABLISH ARRAY SIZES
```

```
C
      DIMENSION C(90), DENSTY(90), ENTHIP(90), OMEGA(90), PRESS(90),
     *           RADIUS(90), THETA(90), U(90), V(90), VEL(90), X(90),
     *           Y(90)
C
C   INITIALIZE ARRAYS TO PREVENT UNWANTED RESULTS
C
      DATA C/90*0.00000000/,DENSTY/90*0.00000000/,ENTHLP/90*0.00000000/
      DATA OMEGA/90*0.00000000/,PRESS/90*0.00000000/
      DATA RADIUS/90*0.00000000/,THETA/90*0.00000000/,U/90*0.00000000/
      DATA V/90*0.00000000/,VEL/90*0.00000000/,X/90*0.00000000/
      DATA Y/90*0.00000000/
C
C   DECLARE CONSTANTS
C
      DATA GAMMA/1.4/,AIRM/28.974/,AIRR/8314.41/
      DATA ARESP/'AIR '/,WRESP/'WATE'/,YES/'YES '/,NO/'NO  '/
C
C
C   INITIALIZE KEY VARIABLES
C
C     BEGIN (PROGRAM CONEFLOW)
C
1     J = 2
      JAND1 = J + 1
      GAMMA1 = (GAMMA - 1.0)/2.0
      GAMMA2 = (GAMMA + 1.0)/(GAMMA * 2.0)
      GAMMA3 = GAMMA/(GAMMA - 1.0)
      STEP = DTOR(1.000000000)
      TVALUE = 0.000001
C
C
C   DETERMINE IF A COMPLETE OR A SUMMARY PRINTOUT IS DESIRED
C
10    WRITE(7,9915)
      READ(7,8902) PRESP
      IF ((PRESP .EQ. YES) .OR. (PRESP .EQ. NO)) GO TO 20
        WRITE(7,9916)
        GO TO 10
20    IF (PRESP .EQ. YES) GO TO 25
        COMPRT = .FALSE.
        GO TO 30
25    COMPRT = .TRUE.
C
C   DETERMINE IF THIS IS AN  -AIR-  OR A  -WATER-  RUN
C
30    WRITE(7,9920)
      READ(7,8902) AWRESP
      IF ((AWRESP .EQ. ARESP) .OR. (AWRESP .EQ. WRESP)) GO TO 35
        WRITE(7,9921)
        GO TO 30
35    IF (AWRESP .EQ. ARESP) GO TO 40
        AIRRUN = .FALSE.
        GO TO 45
40    AIRRUN = .TRUE.
C
C
C   GET VALUES FOR USER SPECIFIED VARIABLES
C
45    WRITE(7,9900)
      READ(7,8900) TEMP1
C
      WRITE(7,9905)
      READ(7,8900) PRESS1
C
      WRITE(7,9910)
      READ(7,8901) SHKANG
C
      IF (AIRRUN) GO TO 50
        WRITE(7,9930)
        READ(7,8901) MFACT
        GO TO 55
C
50    WRITE(7,9925)
      READ(7,8901) MACH1
C
C   VERIFY USER INPUT IS VALID BY CALLING SUBROUTINE CHKINP
C
55    CALL CHKINP(TEMP1,PRESS1,SHKANG,AIRRUN,MACH1,MFACT)
C
C
      OMEGA(2) = DTOR(SHKANG)
60    IF (AIRRUN) GO TO 100
C
C   CALCULATE INITIAL WATER CONDITIONS AT POINTS 1 AND 2
C
      IF (PRESS1 .GT. 0.0) GO TO 65
        PRESS2 = MFACT * 100000000.0
        GO TO 70
65    PRESS2 = PRESS1 * MFACT * 1000.0
70    CALL WSHOCK(TEMP1,PRESS2,VNORM1,VNORM2,C(1),SPVOL2,CREF,C(2),
     *            DENSTY(2),REFVOL,DENSY1,WATSPD)
      VELPS = VNORM1/DSIN(OMEGA(J))
      VTANG = VELPS * DCOS(OMEGA(J))
      MACH1 = VELPS/C(1)
      VEL(2) = DSQRT(VTANG**2 + VNORM2**2)
      MACH2 = VEL(2)/C(2)
C
C     CALCULATE STAGNATION ENTHALPY FOR CALCULATED WATER CONDITIONS
C
      CALL EEV(TEMP1,SPVOL2,PRESS2,ENERG)
      ENTH2 = ENERG + PRESS2 * SPVOL2
      TOTENT = ENTH2 + VEL(2)**2/2.0
90    GO TO 200
C
C  END - CALCULATION OF INITIAL WATER CONDITIONS
C
C  CALCULATE INITIAL AIR CONDITIONS AT POINTS 1 AND 2
C
100   ANGDEF = DEFANG(SHKANG,MACH1)
      RDEFL = RTOD(ANGDEF)
      SHRAD = DTOR(SHKANG)
C
C     CALCULATIONS FOR POINT 1
C
      DENSY1 = PRESS1 * AIRM/(AIRR * TEMP1)
      C(1) = DSQRT(GAMMA * AIRR * TEMP1/AIRM)
      VELFS = C(1) * MACH1
C
C     CALCULATIONS FOR POINT 2
C
      ARGUM1 = (5.+MACH1**2*SIN(SHRAD)**2)/(7.*MACH1**2*SIN(SHRAD)**2
     *-1.0)
      ARGUM2 = (5.+MACH1**2*SIN(SHRAD)**2)*(7.*MACH1**2*SIN(SHRAD)**2
     *-1.)/(36.* MACH1**2*SIN(SHRAD)**2)
C
      MACH2 = SQRT(ARGUM1)/SIN(SHRAD-ANGDEF)
      PRESS2 = PRESS1*((7.*MACH1**2*SIN(SHRAD)**2-1.)/6.)
      TEMP2 = TEMP1 * ARGUM2
      C(2) = C(1) * SQRT(ARGUM2)
      VEL(2) = C(2) * MACH2
      DENSY2 = GAMMA2 * PRESS2/C(2)**2
      VNORM2 = MACH2 * C(2) * SIN(SHRAD-ANGDEF)
      VTANG = VNORM2/TAN(SHRAD-ANGDEF)
C
C     CALCULATE MAXIMUM AIR VELOCITY
C
      MAXVEL = DSQRT(C(2)**2/GAMMA1 + VEL(2)**2)
C
C  END - CALCULATION OF INITIAL AIR CONDITIONS
C
C  CALCULATE BUSEMANN VELOCITY COMPONENTS AT POINT 2
C
200   U(J) = VNORM2 * DSIN(OMEGA(J)) + VTANG * DCOS(OMEGA(J))
```

```
      V(J) = -(VNORM2 * DCOS(OMEGA(J))) + VTANG * DSIN(OMEGA(J))
C
C  CALCULATE THE STREAMLINE ANGLE AT POINT 2
C
      THETA(J) = DATAN(V(J)/U(J))
C
C PRINT INITIAL CONDITIONS AS CALCULATED ABOVE FOR AIR OR WATER
C
300   IF (AIRRUN) GO TO 310
        WRITE(6,9000)
        GO TO 320
C
310   WRITE(6,9003)
320   WRITE(6,9005) TEMP1,PRESS1,SHKANG
C
      IF (AIRRUN) GO TO 330
C
C       PRINT INITIAL WATER CONDITIONS
C
        PKBAR = PRESS2/100000000.0
        WRITE(6,9007)  MFACT
        WRITE(6,9013)
        WRITE(6,9015)  REFVOL, DENSY1
        WRITE(6,9017)  VNORM1, VTANG, VELFS
        WRITE(6,9020)  CREF, C(1)
        WRITE(6,9023)  MACH1
        WRITE(6,9025)
        WRITE(6,9027)  PRESS2, PKBAR, SPVOL2, DENSTY(2)
        WRITE(6,9030)  ENERG, ENTH2, TOTENT
        WRITE(6,9032)  VNORM2, VEL(2), WATSPD
        WRITE(6,9034)  C(2), MACH2
        GO TO 340
C
C     PRINT INITIAL CONDITIONS FOR AIR
C
330   WRITE(6,9023)  MACH1
      WRITE(6,9036)
      WRITE(6,9038)  RDEFL,DENSY1
      WRITE(6,9040)  VELFS,MAXVEL
      WRITE(6,9042)  MACH2, VEL(2), VTANG
C
C     PRINT COMMON INITIAL OUTPUT VARIABLES
C
340   ODEG = RTOD(OMEGA(2))
      TDEG = RTOD(THETA(2))
      WRITE(6,9043)  U(2), V(2)
      WRITE(6,9044)  ODEG, TDEG
C
```

```
C   BEGIN - ITERATION SEQUENCE TO DETERMINE CONE SEMI-VERTEX ANGLE
C
C   CALCULATE RADIUS OF BUSEMANN APPLE CURVE FOR AIR OR WATER
C
400     NUM = VEL(J)  * DSIN(THETA(J))/DSIN(OMEGA(J))
        DEN =  1.0  -  (VEL(J)/C(J))**2 * DSIN(OMEGA(J)  - THETA(J))**2
        RADIUS(J)  = NUM/DEN
C
C   CALCULATE CENTER OF BUSEMANN CIRCLE
C
        X(J)  = U(J)  + RADIUS(J)  * DCOS(OMEGA(J))
        Y(J)  = V(J)  + RADIUS(J)  * DSIN(OMEGA(J))
C
C   DECREMENT THE ANGLE OMEGA
C
        OMEGA(J+1)  = OMEGA(J)  - STEP
C
C   CALCULATE BUSEMANN VELOCITY COMPONENTS AT THE NEXT POINT
C
450     U(J+1)  = X(J)  - RADIUS(J)  * DCOS(OMEGA(J+1))
        V(J+1)  = Y(J)  - RADIUS(J)  * DSIN(OMEGA(J+1))
C
C   CALCULATE THE VELOCITY AT THE NEXT POINT
C
        VEL(J+1)  = DSQRT(V(J+1)**2 + U(J+1)**2)
C
C   CALCULATE THE STREAMLINE ANGLE AT THE NEXT POINT
C
        THETA(J+1)  = DATAN(V(J+1)/U(J+1))
C
C   PRINT RESULTS IF THIS IS A COMPLETE PRINTOUT
C
        IF (.NOT. COMPRT)  GO TO 475
          WRITE(6,9045)  J,RADIUS(J)
          WRITE(6,9050)  X(J),Y(J)
          WRITE(6,9075)
          WRITE(6,9080)  JAND1
          WRITE(6,9060)  JAND1,U(J+1),JAND1,V(J+1)
          WRITE(6,9065)  JAND1,VEL(J+1)
          ODEG = RTOD(OMEGA(J+1))
          TDEG = RTOD(THETA(J+1))
          WRITE(6,9070)  JAND1,ODEG,JAND1,TDEG
C
C   TEST TO DETERMINE IF CONE SURFACE HAS BEEN REACHED
C
475     IF (TVALUE .GT. DABS(THETA(J+1)-OMEGA(J+1)))  GO TO  1000
C
C         ENSURE STREAMLINE ANGLE IS LESS THAN THE ANGLE OMEGA
```

```
C
         IF (THETA(J+1) .LT. OMEGA(J+1)) GO TO 500
           OMEGA(J+1) = DATAN(Y(J)/X(J))
480        GO TO 450
C
C
500    IF (AIRRUN) GO TO 600
C
C   CALCULATE THE WATER CONDITIONS AT THE NEXT POINT
C
505    CALL WATVEL(J,TEMP1,CREF,TOTENT,REFVOL,DENSTY(J),VEL(J+1),C(J+1),
      *             ENTHLP(J+1),PRESS(J+1),DENSTY(J+1))
510    GO TO 650
C
C   CALCULATE THE SPEED OF SOUND IN AIR AT THE NEXT POINT
C
600    C(J+1) = SCSAIR(MAXVEL,VEL(J+1))
C
C   PRINT RESULTS IF A COMPLETE PRINTOUT IS DESIRED
C
650    IF (.NOT. COMPRT) GO TO 700
         IF (AIRRUN) GO TO 675
           PKBAR = PRESS(J+1)/100000000.0
           WRITE(6,9145) JAND1,ENTHLP(J+1)
           WRITE(6,9150) JAND1,PRESS(J+1),PKBAR
           WRITE(6,9155) JAND1,DENSTY(J+1)
675      WRITE(6,9140) JAND1,C(J+1)
C
C   INCREMENT COUNTERS AND REPEAT CALCULATIONS
C
700    J = J + 1
       JAND1 = J + 1
       GO TO 400
C
C
C   THE CONE SURFACE HAS BEEN REACHED - CALCULATE FINAL CONDITIONS
C
1000   IF (AIRRUN) GO TO 1100
C
C     CALCULATE FINAL WATER CONDITIONS AT THE CONE SURFACE
C
       CALL WATVEL(J,TEMP1,CREF,TOTENT,REFVOL,DENSTY(J),VEL(J+1),C(J+1),
      *             ENTHLP(J+1),PRESS(J+1),DENSTY(J+1))
       PRESCS = PRESS(J+1)
       PKBAR = PRESCS/100000000.0
       DENSCS = DENSTY(J+1)
       ENTHCS = ENTHLP(J+1)
       GO TO 1200
```

```
C
C       CALCULATE FINAL AIR CONDITIONS AT THE CONE SURFACE
C
1100    C(J+1) = SCSAIR(MAXVEL,VEL(J+1))
        TEMPCS = C(J+1)**2 * AIRM/(GAMMA * AIRR)
        PRESCS = PRESS2 * (TEMPCS/TEMP2)**GAMMA3
        DENSCS = GAMMA * PRESCS/C(J+1)**2
C
C       CALCULATE CONE CONDITIONS COMMON TO AIR OR WATER CASES
C
1200    DRAGCO = 2.0 * (PRESCS - PRESS1)/(DENSY1 * VELFS**2)
        MACHCS = VEL(J+1)/C(J+1)
        CONANG = RTOD(OMEGA(J+1))
C
C       DISPLAY FINAL RESULTS
C
        WRITE(6,9075)
        WRITE(6,9083)
        WRITE(6,9100)  CONANG
        WRITE(6,9085)  SHKANG
        WRITE(6,9090)  MACH1
        WRITE(6,9095)  VELFS
        WRITE(6,9130)  VEL(J+1)
        WRITE(6,9120)  C(J+1)
        WRITE(6,9135)  MACHCS
        WRITE(6,9105)  PRESCS
C
        IF (AIRRUN) GO TO 1225
          WRITE(6,9106) PKEAF
          WRITE(6,9111) ENTHCS
          GO TO 1250
C
1225    WRITE(6,9110) TEMPCS
C
1250    WRITE(6,9115)  DENSCS
        WRITE(6,9125)  DRAGCO
        WRITE(6,9075)
C
C   DETERMINE IF ANOTHER RUN IS DESIRED
C
1275    WRITE(7,9940)
        READ(7,8902) AGAIN
C
C   VERIFY CORRECT RESPONSE
C
        IF ((AGAIN .EQ. YES) .OR. (AGAIN .EQ. NO)) GO TO 1300
          WRITE(7,9916)
          GO TO 1275
```

```
C
C  RESPONSE IS CORRECT, TAKE ACTION DIRECTED
C
1300  IF (AGAIN .EQ. NO) GO TO 1400
        WRITE(6,9185)
        WRITE(7,9945)
        GO TO 1
C
1400  WRITE(7,9935)
      STOP
C
C     END  PROGRAM CONEFLOW
C
C  FORMAT STATEMENTS
C
C   THE FOLLOWING FORMAT STATEMENTS ARE FOR READING DATA
C
8900  FORMAT(F12.5)
8901  FORMAT(F8.5)
8902  FORMAT(A4)
C
C   THE FOLLOWING FORMAT STATEMENTS ARE FOR WRITING TO THE OUTPUT FILE
C
9000  FORMAT('*** THIS RUN IS FOR  - WATER - ***'//'   INPUT VALUES WERE
     * AS FOLLOWS:'/)
9003  FORMAT('*** THIS RUN IS FOR  - AIR - ***'//'   INPUT VALUES WERE A
     *S FOLLOWS:'/)
9005  FORMAT('UPSTREAM TEMPERATURE = ',F10.5,'  KELVIN'/'UPSTREAM PRESSU
     *RE = ',F20.5,'  PASCALS'/'SHOCK ANGLE = ',F8.4,'  DEGREES')
9007  FORMAT('THE PRESSURE MULTIPLICATION FACTOR (MFACT) = ',F8.5/)
9013  FORMAT('   CONDITIONS CALCULATED FOR POINT 1 UPSTREAM:'/)
9015  FORMAT('REFERENCE SPECIFIC VOLUME = ',F20.10,'  M3/KG'/'DENSITY AT
     * POINT 1 = ',F20.10,'  KG/M3'/)
9017  FORMAT('NORMAL COMPONENT OF VELOCITY = ',F20.10,'  M/S'/'TANGENTIA
     *L COMPONENT OF VELOCITY = ',F20.10,'  M/S'/'FREESTREAM VELOCITY =
     *',F20.10,'  M/S'/)
9020  FORMAT('REFERENCE SPEED OF SOUND = ',F20.10,'  M/S'/'SPEED OF SOUN
     *D AT POINT 1 = ',F20.10,'  M/S'/)
9023  FORMAT('FREESTREAM MACH NUMBER = ',F20.10/)
9025  FORMAT(/'   CONDITIONS CALCULATED FOR POINT 2 DOWNSTREAM'/)
9027  FORMAT('PRESSURE AT POINT 2 = ',F25.10,' PASCALS'/'   THIS PRESSUR
     *E IN KILOBARS = ',F10.5,' KILOBARS'/'SPECIFIC VOLUME AT POINT 2 =
     *',F15.10,'  M3/KG'/'DENSITY AT POINT 2 = ',F20.10,' KG/M3'/)
9030  FORMAT('EBAR (FROM EEV) AT POINT 2 = ',F20.10,'  J/KG '/'ENTHALPY
     *AT POINT 2 = ',F20.10,'  J/KG '/'STAGNATION ENTHALPY = ',F20.10,'
     * J/KG '/)
9032  FORMAT('NORMAL COMPONENT OF VELOCITY = ',F20.10,'  M/S'/'VELOCITY
     *AT POINT 2 = ',F20.10,'  M/S'/'WATER VELOCITY AT POINT 2 = ',F10.4
     *,' M/S'/)
9034  FORMAT('SPEED OF SOUND AT POINT 2 = ',F20.10,'  M/S'/'MACH NUMBER
     *AT POINT 2 = ',F20.10/)
9036  FORMAT('    CALCULATED INITIAL CONDITIONS WERE:'/)
9038  FORMAT('DEFLECTION ANGLE = ',F9.4,'  DEGREES'/'DENSITY AT POINT 1
     *= ',F20.10,'  KG/M3'/)
9040  FORMAT('FREESTREAM VELOCITY = ',F20.10,'  M/S'/'MAXIMUM AIR VELOCI
     *TY = ',F10.5,'  M/S'/)
9042  FORMAT('MACH NUMBER AT POINT 2 = ',F20.10/'VELOCITY AT POINT 2 = '
     *,F20.10,'  M/S'/'TANGENTIAL COMPONENT OF VELOCITY = ',F20.10,'  M/
     *S'/)
9043  FORMAT('X-COMPONENT OF BUSEMANN VELOCITY = ',F25.15,'  M/S'/'Y-COM
     *PONENT OF BUSEMANN VELOCITY = ',F25.15,'  M/S'/)
9044  FORMAT('OMEGA AT POINT 2 = ',F9.4,'  DEGREES'/'STREAMLINE ANGLE AT
     * POINT 2 = ',F9.4,'  DEGREES'/)
9045  FORMAT('BUSEMANN RADIUS AT POINT ',I2,' = ',F25.15,'  M/S')
9050  FORMAT('X-COORDINATE OF BUSEMANN CENTER = ',F25.15,'  M/S'/'Y-COOR
     *DINATE OF BUSEMANN CENTER = ',F25.15,'  M/S'/)
9060  FORMAT('X-COMPONENT OF VELOCITY AT POINT ',I2,' = ',F25.15,'  M/S'
     */'Y-COMPONENT OF VELOCITY AT POINT ',I2,' = ',F25.15,'  M/S')
9065  FORMAT('VELOCITY AT POINT ',I2,' = ',F25.15,'  M/S'/)
9070  FORMAT('OMEGA AT POINT ',I2,' = ',F9.4,'  DEGREES'/'STREAMLINE ANG
     *LE AT POINT ',I2,' = ',F9.4,'  DEGREES'/)
9075  FORMAT(/'* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     * * * * * * * * * * * * * * * * * * * *'/)
9080  FORMAT('    POINT = ',I2/)
9083  FORMAT('CONE HAS BEEN REACHED - FOLLOWING IS A SUMMARY OF RESULTS:
     *'/)
9085  FORMAT('SHOCK ANGLE =  ',F9.4,'  DEGREES')
9090  FORMAT('FREESTREAM MACH NUMBER =  ',F10.5)
9095  FORMAT('FREESTREAM VELOCITY =  ',F15.5,'  M/S'/)
9100  FORMAT('CONE SEMI-VERTEX ANGLE =  ',F9.4,'  DEGREES'/)
9105  FORMAT('PRESSURE AT CONE SURFACE =  ',F25.10,'  PASCALS')
9106  FORMAT('    PRESSURE AT CONE SURFACE = ',F10.5,' KILOBARS')
9110  FORMAT('TEMPERATURE AT CONE SURFACE =  ',F25.15,'  KELVIN')
9111  FORMAT('ENTHALPY AT CONE SURFACE = ',F20.10,' J/KG ')
9115  FORMAT('DENSITY AT CONE SURFACE =  ',F25.15,'  KG/M3')
9120  FORMAT('SPEED OF SOUND AT CONE SURFACE =  ',F25.15,'  M/S')
9125  FORMAT('DRAG COEFFICIENT (CD) =  ',F25.15/)
9130  FORMAT('VELOCITY AT CONE SURFACE =  ',F25.15,'  M/S')
9135  FORMAT('MACH NUMBER AT CONE SURFACE =  ',F25.15/)
9140  FORMAT('SPEED OF SOUND AT POINT ',I2,' = ',F25.15,'  M/S'/)
9145  FORMAT('ENTHALPY AT POINT ',I2,' = ',F25.15,'  J/KG ')
9150  FORMAT('PRESSURE AT POINT ',I2,' = ',F25.10,'  PASCAL'/'   THIS PR
     *ESSURE IN KILOBARS = ',F10.5,' KILOBARS')
9155  FORMAT('DENSITY AT POINT ',I2,' = ',F25.15,'  KG/M3')
9185  FORMAT(/'THIS RUN WAS COMPLETED. NEXT RUN IS BEGINNING.'/'# # # #
     * # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
     * # # # # # # # # # # # # #'///)
C
C   THE FOLLOWING FORMAT STATEMENTS ARE FOR WRITING TO THE TERMINAL
C
9900  FORMAT(/'ENTER THE FREESTREAM TEMPERATURE IN DEGREES KELVIN: ')
9905  FORMAT('ENTER THE FREESTREAM PRESSURE IN PASCALS: ')
9910  FORMAT('ENTER THE SHOCK ANGLE IN DEGREES (E.G. 30.0): ')
9915  FORMAT('DO YOU WANT A COMPLETE PRINTOUT? (YES OR NO)')
9916  FORMAT('ERROR - YOUR ANSWER TO THE QUESTION ABOVE MUST BE EITHER
     ** YES * OR * NO *'/'PLEASE TRY AGAIN.')
9920  FORMAT('IS THIS RUN FOR AIR OR WATER? (ANSWER = AIR OR WATER)')
9921  FORMAT('ERROR - YOUR ANSWER TO THE QUESTION ABOVE MUST BE EITHER
     ** AIR * OR * WATER *'/'PLEASE TRY AGAIN.'/)
9925  FORMAT('ENTER THE FREESTREAM MACH NUMBER (E.G. 3.0):')
9930  FORMAT('ENTER THE KILOBAR MULTIPLICATION FACTOR'/'(NOTE: MFACT MUS
     *T BE GREATER THAN 0.0 BUT LESS THAN 100.1):')
9935  FORMAT('PROGRAM EXECUTION IS COMPLETED. YOU CAN REVIEW THE OUTPUT
     *FROM THIS RUN'/'BY USING THE * BROWSE * OR * XEDIT * COMMANDS.')
9940  FORMAT('WOULD YOU LIKE TO MAKE ANOTHER RUN OF THIS PROGRAM? (YES O
     *R NO)')
9945  FORMAT('VERY WELL. EXECUTION OF NEW RUN BEGINS.'/)
C
      END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                            FUNCTION DTOR
C
C   THIS FUNCTION CONVERTS AN ANGULAR MEASUREMENT, RECEIVED AS AN INPUT
C   PARAMETER TO THE FUNCTION, FROM DEGREES TO THE CORRESPONDING MEASURE
C   IN RADIANS.
C
C   NO GLOBAL VARIABLES ARE AFFECTED BY THIS FUNCTION.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION DTOR(ANGLED)
C
C   DECLARE FUNCTION VARIABLES
C
        DOUBLE PRECISION ANGLED
C
C     BEGIN (FUNCTION DTOR)
C
        DTOR = ANGLED * 3.14159265 35/180.0
        RETURN
C
C     END (FUNCTION DTOR)
C
      END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                          FUNCTION RTOD
C
C  THIS FUNCTION CONVERTS AN ANGULAR MEASUREMENT, RECEIVED AS AN INPUT
C  PARAMETER TO THE FUNCTION, FROM RADIANS TO THE CORRESPONDING MEASURE
C  IN DEGREES.
C
C  NO GLOBAL VARIABLES ARE AFFECTED BY THIS FUNCTION.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION RTOD(ANGLER)
C
C  DECLARE FUNCTION VARIABLES
C
        DOUBLE PRECISION ANGLER
C
C     BEGIN (FUNCTION RTOD)
C
        RTOD = ANGLER * 180.0/3.1415926535
        RETURN
C
C     END (FUNCTION RTOD)
C
      END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                         FUNCTION DEFANG
C
C  THIS FUNCTION CALCULATES THE DEFLECTION ANGLE WHICH RESULTS WHEN A
C  STREAMLINE PASSES ACROSS AN OBLIQUE SHOCK FRONT IN AIR. THE FUNCTION
C  RECEIVES THE FOLLOWING MAIN PROGRAM VARIABLES AS INPUT PARAMETERS:
C    SHKANG AND MACH1
C  THE DEFLECTION ANGLE IS COMPUTED AND IS RETURNED IN RADIAN MEASURE.
```

```
C
C   NOTE: THE FUNCTION HAS A FEATURE WHICH TESTS TO DETERMINE IF THE DE-
C         FLECTION ANGLE HAS BECOME LARGER THAN 1 RADIAN. IF IT HAS, THE
C         MAIN PROGRAM'S EXECUTION IS STOPPED DUE TO A FATAL ERROR. IN
C         ACCORDANCE WITH KINNEY AND GRAHAM'S, EXPLOSIVE SHOCKS IN AIR,
C         THE DEFLECTION ANGLE CAN NEVER BE LARGER THAN 1 RADIAN.
C
C   NO GLOBAL VARIABLES ARE AFFECTED BY THIS SUBROUTINE.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION DEFANG(SANG,MUP)
C
C      DECLARE FUNCTION VARIABLES
C
        REAL ANG,DDEN,DLHS,DNUM,G,MUP,PANG,RHS,SANG,SRAD,TESTV
C
C     BEGIN (FUNCTION DEFANG)
C
C   INITIALIZE KEY VARIABLES
C
        DATA G/1.4/
        TESTV = 0.00001
        ANG = 0.000001
        SRAD = DTOR(SANG)
C
C   COMPUTE THE RIGHT-HAND SIDE OF THE EQUATION
C
        DNUM = (2.+(G-1.)*MUP**2*SIN(SRAD)**2)*TAN(SRAD)
        DDEN = (G+1.)*MUP**2*SIN(SRAD)**2
        RHS = DNUM/DDEN
C
C   ITERATE THE LEFT-HAND SIDE OF THE EQUATION TO FIND ANGDEF
C
10      DLHS = TAN(SRAD-ANG)
        IF (TESTV .GE. ABS(RHS-DLHS)) GO TO 30
          ANG = ANG + 0.0000001
C
C           IF ANGDEF IS GREATER THAN 1 RADIAN, THERE IS A FATAL ERROR.
C
          IF (ANG .LE. 1.0) GO TO 20
            WRITE(6,50)
            WRITE(7,50)
            STOP
C
20        GO TO 10
30      DEFANG = ANG
        RETURN
```

```
C
C     END (FUNCTION DEFANG)
C
C  FORMAT STATEMENTS
C
50    FORMAT(/'A FATAL ERROR HAS OCCURRED IN FUNCTION DEFANG - THE ANGLE
     * HAS BECOME LARGER'/'THAN 1 RADIAN. THIS IS PHYSICALLY IMPOSSIBLE.
     * PROGRAM EXECUTION HAS TERMINATED.'/)
C
      END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                        FUNCTION SOSAIR
C
C  THIS FUNCTION CALCULATES THE SPEED OF SOUND AT ANY POINT IN AN AIR
C  ENVIRONMENT. THE FUNCTION RECEIVES THE FOLLOWING MAIN PROGRAM VAR-
C  IABLES AS INPUT PARAMETERS:
C    MAXVEL AND VEL(J+1)
C
C  NO GLOBAL VARIABLES ARE AFFECTED BY THIS FUNCTION.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      FUNCTION SOSAIR(VMAX,SVEL)
C
C  DECLARE FUNCTION VARIABLES
C
        REAL G1
        DOUBLE PRECISION VMAX,SVEL
C
C  DECLARE CONSTANTS
C
        DATA G1/0.2/
C
C     BEGIN (FUNCTION SOSAIR)
C
        SOSAIR = DSQRT(G1 * (VMAX**2 - SVEL** 2))
        RETURN
C
C     END (FUNCTION SOSAIR)
C
      END
C
C
```

```
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                          SUBROUTINE CHKINP
C
C   THIS SUBROUTINE DETERMINES IF THE INPUT VALUES PROVIDED TO THE MAIN
C   PROGRAM BY THE USER FALL WITHIN ALLOWED RANGES. THE SUBROUTINE RE-
C   CEIVES THE FOLLOWING MAIN PROGRAM VARIABLES AS INPUT PARAMETERS:
C     TEMP1, PRESS1, SHKANG, AIRRUN, AND EITHER MACH1 OR MFACT.
C
C   IF ALL VARIABLES ARE WITHIN ALLOWABLE RANGE, THE SUBROUTINE DOES NOT
C   CHANGE ANY VARIABLE. IF, HOWEVER, A VARIABLE IS OUT OF RANGE, THE
C   SUBROUTINE REPEATEDLY REQUESTS NEW INPUT FOR THAT VARIABLE FROM THE
C   USER UNTIL THAT VARIABLE IS WITHIN SPECIFICATION. IN THIS CASE, THE
C   VARIABLE CHANGED IS RETURNED TO THE MAIN PROGRAM AS AN OUTPUT PARA-
C   METER.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE CHKINP(TANS,PANS,SANS,AANS,MANS,MFANS)
C
C   DECLARE SUBROUTINE VARIABLES
C
        REAL TANS,PANS,SANS,MANS,MFANS,MINANG,RSANS
        LOGICAL AANS
C
C       BEGIN (SUBROUTINE CHKINP)
C
10      IF ((SANS .GT. 0.0) .AND. (SANS .LT. 90.0)) GO TO 15
          WRITE(7,9000)
          WRITE(7,9005)
          READ(7,8950) SANS
          GO TO 10
C
C   DETERMINE WHETHER THIS IS AN -AIR- OR A -WATER- RUN
C
15      IF (AANS) GO TO 100
C
C   FOLLOWING TESTS ARE CONDUCTED FOR A -WATER- RUN
C
20        IF ((TANS .GE. 273.16) .AND. (TANS .LE. 373.16)) GO TO 30
            WRITE(7,9010)
            WRITE(7,9015)
            READ(7,8955) TANS
            GO TO 20
C
C                     NOTE: 10000000000.0 PASCALS = 100.0 KBARS
```

```
C
30         IF ((PANS .GE. 0.0) .AND. (PANS .LE. 1000000000.0)) GO TO 40
             WRITE(7,9020)
             WRITE(7,9025)
             READ(7,8955) PANS
             GO TO 30
C
40         IF ((MFANS .GT. 0.0) .AND. (MFANS . LE. 100.0)) GO TO 50
             WRITE(7,9030)
             WRITE(7,9035)
             READ(7,8950) MFANS
             GO TO 40
C
C   ENSURE PRESSURE AT POINT 2 IS LESS THAN 100.0 KBARS
C
50         IF ((PANS*MFANS) .LE. 1000000000.0) GO TO 60
             WRITE(7,9040)
             WRITE(7,9035)
             READ(7,8950) MFANS
             GO TO 40
C
C   ALL INPUT VALUES FOR A -WATER- RUN ARE WITHIN LIMITS
C
60         WRITE(7,9045)
           GO TO 200
C
C   FOLLOWING TESTS ARE CONDUCTED FOR AN -AIR- RUN
C
100      IF (TANS .GT. 0.0) GO TO 110
           WRITE(7,9050)
           WRITE(7,9055)
           READ(7,8955) TANS
           GO TO 100
C
110      IF (EANS .GT. 0.0) GO TO 120
           WRITE(7,9060)
           WRITE(7,9065)
           READ(7,8955) EANS
           GO TO 110
C
120      IF (MANS .GT. 1.0) GO TO 130
           WRITE(7,9070)
           WRITE(7,9075)
           READ(7,8950) MANS
C
C   ENSURE SHOCK ANGLE IS ABOVE THE MINIMUM FOR THE MACH NUMBER GIVEN
C
130      MINANG = ARSIN(1.0/MANS)
```

```
          RSANS = DTOR(SANS)
          IF (RSANS .GE. MINANG) GO TO 140
            WRITE(7,9085)
            WRITE(7,9005)
            READ(7,8950) SANS
            WRITE(7,9075)
            WRITE(7,8950)  MANS
            GO TO 10
C
C   ALL VALUES ARE WITHIN LIMITS FOR AN -AIR- RUN
C
140       WRITE(7,9080)
C
C
200     RETURN
C
C       END (SUBROUTINE CHKINP)
C
C   FORMAT STATEMENTS
C
C      READ FORMAT STATEMENTS
C
8950    FORMAT(F8.5)
8955    FORMAT(F12.5)
C
C      WRITE FORMAT STATEMENTS
C
9000    FORMAT(/'ERROR - THE SHOCK ANGLE MUST BE GREATER THAN 0.0 BUT LESS
     *  THAN 90.0')
9005    FORMAT('PLEASE RE-ENTER THE SHOCK ANGLE IN DEGREES (E.G. 30.0):')
9010    FORMAT(/'ERROR - THE TEMPERATURE MUST BE GREATER THAN 273.16 KELVI
     *N BUT LESS THAN'/'373.16 KELVIN.')
9015    FORMAT('PLEASE RE-ENTER THE FREESTREAM TEMPERATURE IN DEGREES KELV
     *IN:')
9020    FORMAT(/'ERROR - THE PRESSURE MUST BE GREATER THAN 0.0 PASCALS '/'
     *BUT LESS THAN 10130000000.0 PASCALS.')
9025    FORMAT('PLEASE RE-ENTER THE FREESTREAM PRESSURE IN PASCALS:')
9030    FORMAT(/'ERROR - THE KILOBAR MULTIPLICATION FACTOR MUST BE GREATER
     *  THAN 0.0'/'BUT LESS THAN 100.1')
9035    FORMAT('PLEASE RE-ENTER THE KILOBAR MULTIPLICATION FACTOR:')
9040    FORMAT('THE PRESSURE AT POINT 2 WILL BE OUTSIDE THE LIMITS FOR THI
     *S PROGRAM.'/'THE COMBINATION OF FREESTREAM PRESSURE TIMES THE MULT
     *IPLICATION FACTOR MUST'/'BE LESS THAN 100.1 KILOBARS.  YOU MUST EN
     *TER A MULTIPLICATION FACTOR WHICH'/'WILL BRING THE PRESSURE AT POI
     *NT 2 WITHIN LIMITS.'/)
9045    FORMAT(/'ALL INPUT VALUES FOR THE CALCULATIONS IN * WATER * ARE WI
     *THIN LIMITS.'/'EXECUTION OF YOUR PROGRAM WILL NOW BEGIN:'/)
9050    FORMAT(/'ERROR - THE FREESTREAM AIR TEMPERATURE MUST BE GREATER TH
     *AN 0.0.')
9055  FORMAT('PLEASE RE-ENTER THE FREESTREAM TEMPERATURE IN DEGREES KELV
     *IN:')
9060  FORMAT(/'ERROR - THE FREESTREAM AIR PRESSURE MUST BE GREATER THAN
     *0.0.')
9065  FORMAT('PLEASE RE-ENTER THE FREESTREAM PRESSURE IN PASCALS:')
9070  FORMAT(/'ERROR - THE FREESTREAM MACH NUMBER MUST BE GREATER THAN 1
     *.0.')
9075  FORMAT('PLEASE RE-ENTER THE FREESTREAM MACH NUMBER:')
9080  FORMAT(/'ALL INPUT VALUES FOR THE CALCULATIONS IN * AIR * ARE WITH
     *IN LIMITS.'/'EXECUTION OF YOUR PROGRAM WILL NOW BEGIN:'/)
9085  FORMAT('ERROR - YOUR COMBINATION OF MACH NUMBER AND SHOCK ANGLE IS
     *IS NOT POSSIBLE.'/'THE SHOCK ANGLE MUST BE GREATER THAN ARCSIN(1.0
     */M1). PLEASE ENTER'/'NEW VALUES FOR THE SHOCK ANGLE AND UPSTREAM M
     *ACH NUMBER (ADJUST 1 OR BOTH AS NEEDED).')
C
      END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                           SUBROUTINE WSHOCK
C
C  THIS SUBROUTINE COMPUTES THE THERMODYNAMIC PROPERTIES OF WATER AT
C  POINT 1 UPSTREAM OF A SHOCK FRONT AND AT POINT 2 DOWNSTREAM OF A
C  SHOCK FRONT. THE SUBROUTINE RECEIVES THE FOLLOWING MAIN PROGRAM VAR-
C  IABLES AS INPUT PARAMETERS:
C    TEMP1 AND PRESS2
C
C  THE FOLLOWING MAIN PROGRAM VARIABLES ARE COMPUTED BY THIS SUBROUTINE
C  AND ARE RETURNED AS OUTPUT PARAMETERS:
C    VNORM1, VNORM2, C(1), SPVOL2, CREF, C(2) , DENSTY(2) , REFVOL, DENSY1
C
C  THIS SUBROUTINE DOES NOT UTILIZE ANY OTHER SUBROUTINE OR FUNCTION.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE WSHOCK(WTEMP1,WSPRES,VEL1,VEL2,SCS1,SVOL2,REFSCS,SCS2,
     *                  DENS2,SPVOLO,DENS1,WVEL2)
C
C  DECLARE SUBROUTINE VARIABLES
C
        REAL N, N1, N2, N3, N4, WSTEST
C
        DOUBLE PRECISION B6, B7, BXV, ISPVOL, ITEMP1, ITEMP2, ITEMP3,
     *                   SLOPE1, SLOPE2, SPVOLO, SPVOL1, SVRAT, VEL1,
     *                      VEL2, WSENT2, WSENT3, WSENT4, WSENT5, WSENT6,
     *                      WSENT7, WSENT8, WSENT9, WSPRES, WSTEMP, WTEMP1
C
          INTEGER COUNT1, COUNT2
C
C   DECLARE CONSTANTS
C
          DATA N/7.15/,WSTEST/0.00001/
C
C   INITIALIZE KEY VARIABLES
C
          ITEMP2 = 200.0
          N1 = (N + 1.0)/(N - 1.0)
          N2 = (N - 1.0)/N
          N3 = -(1.0/N)
          N4 = (N - 1.0)/(2.0 * N)
          COUNT1 = 0
          COUNT2 = 0
C
C      BEGIN (SUBROUTINE WSHOCK)
C
          ITEMP1 = WTEMP1 - 273.16
          WSTEMP = ITEMP1
          SPVOL1 = (0.99415 + 0.0002929 * (WTEMP1 - 298.16) + 0.000003241
     *        * (WTEMP1 - 298.16)**2)/1000.0
C
4050      ISPVOL = (0.99415 + 0.0002929 * (WSTEMP - 25.0) + 0.000003241 *
     *        (WSTEMP - 25.0)**2)/1000.0
          COUNT2 = COUNT2 + 1
          SPVOL0 = ISPVOL
          B7 = 101300000.0 * (3.134 - 0.00165 * (WSTEMP - 55.0) -
     *        0.0001181 * (WSTEMP - 55.0)**2 + 0.000000532 * (WSTEMP -
     *        55.0)**3)
C
          IF (COUNT2 .GT. 1) GO TO 4100
            B6 = B7
C
4100      BXV = B7 * ISPVOL
          WSENT9 = 3.9644 * (WSTEMP - WTEMP1 + 273.16) + 0.000312 * (WSTEM
     *P**2 - (WTEMP1 - 273.16)**2)
          SVRAT = WSPRES/B7 + 1.0
          WSENT8 = BXV/2.0 * (SVRAT - N1 * (SVRAT**N2 - 1.0) - SVRAT**N3 -
     *          (ISPVOL - SPVOL1) * (SVRAT - 1.0)/ISPVOL)/1000.0
C
          IF (COUNT1 .GT. 0) GO TO 4200
            WSENT2 = WSENT8
            WSENT3 = WSENT9
            WSTEMP = ITEMP2
            COUNT1 = COUNT1 + 1
            GO TO 4050
C
4200      IF (COUNT1 .GT. 1) GO TO 4300
            WSENT4 = WSENT8
            WSENT5 = WSENT9
            COUNT1 = COUNT1 + 1
C
4250        SLOPE1 = (WSENT5 - WSENT3)/(ITEMP2 - ITEMP1)
            SLOPE2 = (WSENT4 - WSENT2)/(ITEMP2 - ITEMP1)
            ITEMP3 = ITEMP1 + (WSENT2 - WSENT3) /(SLOPE1 - SLOPE2)
            WSTEMP = ITEMP3
            GO TO 4050
C
4300      WSENT6 = WSENT8
          WSENT7 = WSENT9
C
          IF (WSTEST .GT.DABS(WSENT6 - WSENT7))  GO TO 4400
            ITEMP1 = ITEMP2
            ITEMP2 = ITEMP3
C
            IF (ITEMP1 .EQ. ITEMP2) GO TO 4400
              WSENT2 = WSENT4
              WSENT4 = WSENT6
              WSENT3 = WSENT5
              WSENT5 = WSENT7
              GO TO 4250
C
4400      DENS1 = 1.0/SPVOL1
          SVOL2 = SPVOL0 * SVRAT**N3
          WVEL2 = DSQRT(WSPRES * (SPVOL1 - SVOL2))
          VEL1 = WSPRES * SPVOL1/WVEL2
          REFSOS = DSQRT(EXV * N)
          SOS2 = REFSOS * SVRAT**N4
          SOS1 = DSQRT(B6 * N * SPVOL1)
          VEL2 = VEL1 - WVEL2
          DENS2 = 1.0/SVOL2
          RETURN
C
C        END (SUBROUTINE WSHOCK)
C
         END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

```
C                          SUBROUTINE WATVEL
C
C  THIS SUBROUTINE CALCULATES THE THERMODYNAMIC PROPERTIES OF WATER AT
C  ANY GIVEN POINT IN THE WATER. THE SUBROUTINE RECEIVES THE FOLLOWING
C  MAIN PROGRAM VARIABLES AS INPUT PARAMETERS:
C    J, TEMP1, CREF, TOTENT, REFVOL, DENSTY(J), VEL(J+1)
C
C  THE FOLLOWING MAIN PROGRAM VARIABLES ARE COMPUTED BY THIS SUBROUTINE
C  AND ARE RETURNED AS OUTPUT PARAMETERS:
C    C(J+1), ENTHLP(J+1), PRESS(J+1), DENSTY(J+1)
C
C  THIS SUBROUTINE UTILIZES THE SUBROUTINE EPV IN ITS CALCULATIONS.
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE WATVEL(K,TEM1,REFSPD,SENTH,RVCI,DENSJ,VELJ,SOSJ1,ENTHAL
     *                 ,PRESJ1,DENSJ1)
C
C  DECLARE SUBROUTINE VARIABLES
C
        REAL N,N1,WTEST
C
        DOUBLE PRECISION B, DENS3, DENS4, DENS5, DENSJ, DENSJ1, ENTHAL,
     *                   ENTHP1, ENTHP2, ENTHP3, PRESJ1, REFSPD, RVCI,
     *                   SENTH, SOSJ1, TEM1, VELJ, WDENS, WENERG, WENTH,
     *                   WPRESS, WSPVOL, WTEMP
C
        INTEGER COUNT
C
C  INITIALIZE KEY VARIABLES
C
        WTEST = 10.0
        N = 7.15
        N1 = (N - 1.0)/2.0
        COUNT = 0
        WTEMP = 0.0
C
C  CALCULATE INITIAL VALUES FOR KEY VARIABLES
C
        ENTHAL = SENTH - VELJ**2/2.0
        DENS3 = DENSJ
        DENS4 = 1.02 * DENS3
        WDENS = DENS3
        B = 10130000.0 * (3.134 - 0.00165 * (WTEMP - 55.0) - 0.0001181
     *      * (WTEMP - 55.0)**2 + 0.000000532 * (WTEMP - 55.0)**3)
C
C     BEGIN (SUBROUTINE WATVEL)
C
```

```
C  CALCULATE PRESSURE USING THE MODIFIED TAIT  EQUATION
C
2050    WPRESS = E * ((WDENS * RVOL)**N - 1.0)
        WSPVCL = 1.0/WDENS
C
C   CALCULATE THE WATER'S THERMODYNAMIC PROPERTIES
C
        CALL EPV(TEM1,WSPVCL,WPRESS,WENERG)
        WENTH = WENERG + WPRESS * WSPVCL
C
        IF (COUNT .GT. 0) GO TO 2100
          ENTHP1 = WENTH
          WDENS = DENS4
          COUNT = COUNT + 1
          GO TO 2050
C
2100    IF (COUNT .GT. 1) GO TO 2200
          ENTHP2 = WENTH
          COUNT = COUNT + 1
2150      DENS5 = DENS3 + (ENTHAL - ENTHP1) * (DENS4 - DENS3)/(ENTHP2 -
     *            ENTHP1)
          WDENS = DENS5
          GO TO 2050
C
2200    ENTHP3 = WENTH
C
        IF (WTEST .GT. DABS(ENTHAL - ENTHP3)) GO TO 2300
          DENS3 = DENS4
          DENS4 = DENS5
          ENTHP1 = ENTHP2
          ENTHP2 = ENTHP3
          GO TO 2150
C
2300    PRESJ1 = WPRESS
        DENSJ1 = WDENS
        SOSJ1 = REFSPD * (DENS5 * RVOL)**N1
        RETURN
C
C     END (SUBROUTINE WATVEL)
C
      END
C
C
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                             SUBROUTINE EPV
```

```
C
C   THIS SUBROUTINE IS INCLUDED FOR THE INFORMATIONAL PURPOSE OF SHOWING
C   THE USER HOW THE PROGRAM WRITTEN BY A.E. FUHS WAS CONVERTED TO THE
C   FORTRAN LANGUAGE. THE FOLLOWING IS A DIRECT TRANSLATION OF THAT WORK.
C
C   THE SUBROUTINE IS DESCRIBED AND FLOWCHARTED IN NAVAL POSTGRADUATE
C   SCHOOL REPORT NPS67-82-001 BY A.E. FUHS. IT IS USED IN THIS WORK BY
C   PERMISSION OF THE AUTHOR.
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SUBROUTINE EPV(T1EPV,VEPV,PEPV,E3)
C
C   DECLARE PROGRAM VARIABLES
C
        REAL LOOP7,N,N5,N6,N7
C
        DOUBLE PRECISION X1,X2,X3,X4,Y1,Y2,Y3,B,V8,P9,Z1,CC,E1,E2,M6,
     *                   YDIF,XDIF,TEPV,T1EPV,VEPV,PEPV,E3
C
        INTEGER J2,K1
C
C   INITIALIZE KEY VARIABLES
C
        N = 7.15
        LOOP7 = 0.000001
        X1 = 0.0
        X2 = 144.0
        J2 = 0
        K1 = 0
        TEPV = X1
C
C     BEGIN (SUBROUTINE EPV)
C
        X4 = T1EPV - 273.16
3000    B = 10130000.0 * (3.134 - 0.00165 * (TEPV - 55.0) - 0.0001181 *
     *      (TEPV - 55.0)**2 + 0.000000532 * (TEPV - 55.0)**3)
        V8 = (0.99415 + 0.0002929 * (TEPV - 25.0) + 0.00003241 *
     *       (TEPV - 25.0)**2)/1000.0
        P9 = B * ((V8/VEPV)**N - 1.0)
        K1 = K1 + 1
C
        IF (J2 .EQ. 0) GO TO 3100
          IF (J2 .GT. 1) GO TO 3200
            Y2 = P9
            J2 = J2 + 1
            GO TO 3150
```

```
3100    Y1 = P9
        TEPV = X2
        J2 = J2 + 1
        GO TO 3000
C
3150    M6 = (Y2 - Y1)/(X2 - X1)
        X3 = X1 + (PEPV - Y1)/M6
        TEPV = X3
        GO TO 3000
C
3200    Y3 = P9
C
        IF (LOOP7 .GT. DABS((Y3 - PEPV)/PEPV) ) GO TO 3300
          X1 = X2
          X2 = X3
          Y1 = Y2
          Y2 = Y3
          GO TO 3150
C
C  CALCULATE OMEGA OR THE Z-TERM
C
3300    Z1 = V8/VEPV
        CO = DSQRT(N * V8 * B)
        N5 = N - 1.0
        N6 = 1.0/(N * N5)
        N7 = -(1.0/N5)
        E2 = CO**2 * (N6 * Z1**N5 + 1.0/(N * Z1) + N7)
C
C  CALCULATE H10 TERM
C
        E1 = (3.9644 * (X3 - X4) + 0.000312 * (X3**2 - X4**2)) * 1000.0
        E3 = E1 + E2
        RETURN
C
C     END (SUBROUTINE EEV)
C
      END
```

LIST OF REFERENCES

1. G. K. Hartmann and E. G. Hill, Editors, *Underwater Explosion Research, A Compendium of British and American Reports*, Volume I, "The Shock Wave," Office of Naval Research, 1950.

2. R. H. Cole, *Underwater Explosions*, Princeton University Press, 1948.

3. J. M. Richardson, A. B. Arons, and R. R. Halverson, "Hydrodynamic Properties of Sea Water at the Front of a Shock Wave," *Journal of Chemical Physics*, Volume 15, pp. 758-794. 1947.

4. Naval Postgraduate School Report NPS67-82-001, *Computer Program for Calculating Hydrodynamic Properties of Shock Waves in Sea Water*, by A. E. Fuhs, 1 February 1982.

5. G. F. Kinney and K. J. Graham, *Explosive Shocks in Air*, 2nd Edition, Naval Postgraduate School Press, 1983.

6. M. Holt and J. Flores, "Shock Interactions with the Ocean Surface," *Physics of Fluids*, v. 25(2), February 1982.

7. J. S. Rowlinson, *Liquids and Liquid Mixtures*, 2nd Edition, Plenum Press, 1969.

8. A. H. Shapiro, *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Volume II, The Ronald Press Company, 1954.

9. B. J. MacLennan, *Principles of Programming Languages: Design, Evaluation and Implementation*, Holt, Rinehart and Winston, 1983.

10. Z. Kopal, *Tables of Supersonic Flow Around Cones*, Massachusetts Institute of Technology, 1947.

11. H. W. Liepmann and A. Roshko, *Elements of Gas Dynamics*, Wiley and Sons, 1957.

12. R. Franke and G. Nielson, "Smooth Interpolation of Large Sets of Scattered Data," *International Journal for Numerical Methods in Engineering*, v. 15, 1980.

BIBLIOGRAPHY

Integrated Software Systems Corporation, *Display Integrated Software System and Plotting Language (DISSPLA) User's Manual,* 1981.

Moore, John B. and Leo J. Makela, *Structured FORTRAN with WATFIV,* Reston Publishing Company, 1981.

White, Frank M., *Fluid Mechanics,* McGraw-Hill, 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Professor Allen E. Fuhs, Code 67Fu Department of Aeronautics Naval Postgraduate School Monterey, California 93943	2
4.	LT Patrick W. Hughes 15317 Edgehill Drive Dumfries, Virginia 22026	3
5.	Mr. Michael P. Hughes 4180 U.S. 23 North Rogers City, Michigan 49779	1
6.	Mrs. Herbert J. Birchman 7515 28th Avenue N.E. Seattle, Washington 98115	1
7.	Defense Advanced Research Projects Agency DSO, Dr. W. Snowden 1400 Wilson Boulevard Arlington, Virginia 22209	1
8.	Chief of Naval Materiel MAT-0716, A. Faulstich Washington, D.C. 20350	1
9.	Commander, Naval Sea Systems Command PMS-406, R. Whitman Washington, D.C. 20361	1
10.	Commander, Naval Sea Systems Command SEA-62R3, F. Romano Washington, D.C. 20361	1
11.	Commander, David W. Taylor Naval Ship R&D Center Code 175, J. Sykes Bethesda, Maryland 20084	1
12.	Dyna East Corporation Dr. Pei Chi Chou 227 Hemlock Road Wynnewood, Pennslyvania 19096	1
13.	Battelle Columbus Laboratories Dr. B. D. Trott 505 King Street Columbus, Ohio 43201	1

14. D. R. Kennedy Associates, Inc. 1
4940 El Camino Real, Suite 209
P.O. Box 4003
Mountain View, California 94040

15. Zernow Technical Services, Inc. 1
425 West Bonita Avenue, Suite 208
San Dimas, California 91773

16. S-Cubed 1
Dr. R. Sedgwick
P.O. Box 1620
La Jolla, California 92038

17. Honeywell, Inc. U.S.D. 1
Phil Kilpatrick, MN 11-2360
600 Second Street, N.E.
Hopkins, Minnesota 55343

18. Systems Planning Corporation 1
Tom Hafer, 6th Floor
1500 Wilson Boulevard
Arlington, Virginia 22209

19. Lawrence Livermore National Laboratory 1
Milton Finger, Mail Code 324
P.O. Box 808
Livermore, California 94550

20. Los Alamos National Laboratory 1
C. W. Mautz, J960
P.O. Box 1663
Los Alamos, New Mexico 87545

21. Officer in Charge 1
White Oak Laboratory
Naval Surface Weapons Center Detachment
Attn: Mr. Donald Phillips, Code R10A
Silver Spring, Maryland 20910

135375

210259

Thesis
H85783 Hughes
c.1 A computer program to calculate the supersonic flow over a solid cone in air or water.

210259

Thesis
H85783 Hughes
c.1 A computer program to calculate the supersonic flow over a solid cone in air or water.

thesH85783
A computer program to calculate the supe
3 2768 002 13226 8
DUDLEY KNOX LIBRARY